CSE 474/574: Introduction to Machine Learning (Fall 2019)

Sargur N. Srihari
University at Buffalo, The State University of New York
Buffalo, New York 14260
Contact: 716-645-6162 (O), srihari@buffalo.edu

September 9, 2019

1 Task

The task of this project is to perform classification using machine learning. It is for a two class problem. The features used for classification are pre-computed from images of a fine needle aspirate (FNA) of a breast mass. Your task is to classify suspected FNA cells to Benign (class 0) or Malignant (class 1) using logistic regression as the classifier. The dataset in use is the Wisconsin Diagnostic Breast Cancer (wdbc.dataset). The code should be written in Python from scratch. Deadline to submit the code and the report on timberlake server is September 25, 2019.

2 Dataset

Wisconsin Diagnostic Breast Cancer (WDBC) dataset will be used for training, validation and testing. The dataset contains 569 instances with 32 attributes (ID, diagnosis (B/M), 30 real-valued input features). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Computed features describes the following characteristics of the cell nuclei present in the image:

1	radius (mean of distances from center to points on the perimeter)
2	texture (standard deviation of gray-scale values)
3	perimeter
4	area
5	smoothness (local variation in radius lengths)
6	compactness ($perimeter^2/area - 1.0$)
7	concavity (severity of concave portions of the contour)
8	concave points (number of concave portions of the contour)
9	symmetry
10	fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

3 Plan of Work

1. **Extract features values and Image Ids from the data:** Process the original CSV data files into a Numpy matrix or Pandas Dataframe.
2. **Data Partitioning:** Partition your data into training, validation and testing data. Randomly choose 80% of the data for training and the rest for validation and testing.
3. **Train using Logistic Regression:** Use Gradient Descent for logistic regression to train the model using a group of hyperparameters.
4. **Tune hyper-parameters:** Validate the regression performance of your model on the validation set. Change your hyper-parameters. Try to find what values those hyper-parameters should take so as to give better performance on the validation set.
5. **Test your machine learning scheme on the testing set:** After finishing all the above steps, fix your hyper-parameters and model parameter and test your models performance on the testing set. This shows the ultimate effectiveness of your models generalization power gained by learning.

4 Evaluation

1. Print out a graph showing training accuracy versus number of epochs.
2. Evaluate your solution on the test set using Accuracy, Precision and Recall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

5 Deliverables

There are two deliverables: report and code. After finishing the project, you may be asked to demonstrate it to the TAs, particularly if your results and reasoning in your report are not clear enough.

1. Report (30 points)

The report should describe your results, experimental setup and comparison between the results obtained from different setting of the algorithm and dataset. Submit the PDF on a CSE student server with the following script:

`submit_cse474 proj1.pdf` for undergraduates

`submit_cse574 proj1.pdf` for graduates

2. Code (70 points)

   The code for your implementation should be in Python only. You can submit multiple files, but the name of the entrance file should be main.ipynb. Please provide necessary comments in the code. Python code and data files should be packed in a ZIP file named `proj1code.zip`. Submit the Python code on a CSE student server with the following script:

   `submit_cse474 proj1code.zip` for undergraduates

   `submit_cse574 proj1code.zip` for graduates