CSE 40625 — Machine Learning
Assignment 2 (10 points)
Due Date: Feb. 16, 2017 (Sakai Drop Box)

Logistic Regression

Overview

The purpose of this assignment is for you to implement logistic regression, a widely used machine learning model. Logistic regression is a supervised linear classifier that makes predictions by applying the sigmoid function to a linear output. Materials for the assignment, including the dataset, expected output, and template code can be found here on GitHub.

You may use Python libraries for handling data preprocessing and visualization, including but not limited to NumPy, SciPy, pandas, and Matplotlib, but *you may NOT use any Python libraries that employ machine learning models, including but not limited to scikit-learn, StatsModels, TensorFlow, or Orange*. Your solution to the assignment should be individually submitted.

Dataset

You will use a modified version of the “digits” dataset on handwritten digit classification with 2 classes (labeled +1 and −1) for this assignment. The data is provided in comma-separated (CSV) file format. For all rows, the last column designates the class (y) and the remaining columns designate features (X). The first row consists of the feature and class names.

Procedure

Use the modified digits dataset discussed above as the input to a logistic regression model. Insert the threshold into the weight vector as a negative bias term (i.e., feature column of −1). Initialize the weight vector with random samples from a Normal distribution with zero mean and unit variance. Iteratively update the weight vector using the minimum gradient of the error function with a learning rate of 0.01 for at most 500 iterations. Use a probability of 0.5 as the threshold for class prediction. Use the provided code template to guide your thinking.

Output

Your code should output a new line for each correction, showing the current line number (starting from 0), a space, and the current fraction of correctly classified instances to three decimal places. Perform up to 500 iterations of gradient updates, stopping if and when error reaches 0. Note that the format of your output should match the example output provided.

Example output:

```
0 0.598
1 0.603
…
```

Submission

Please submit a Python executable (logistic.py) file of your code to your Sakai Drop Box.

Should you run into any problems, please feel free to email or meet with the instructor.