CS534 — Implementation Assignment 2 — Due 11:59PM Oct 21st, 2020

General instructions.

1. Please use Python 3 (preferably version 3.6+). You may use packages: Numpy, Pandas, and matplotlib, along with any from the standard library (such as 'math', 'os', or 'random' - for example).

2. You should complete this assignment alone. Please do not share code with other students, or copy program files/structure from any outside sources like Github. Your work should be your own.

3. Your source code and report will be submitted through Canvas.

4. You need to follow the submission instructions for file organization (located at the end of the report).

5. Please run your code before submission on one of the OSU EECS servers (i.e. babylon01.eecs.oregonstate.edu). You can make your own virtual environment with the packages we've listed in either your user directory or on the scratch directory. If you're unfamiliar with any of this process, or have limited access, please contact one of the TA's.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. In particular, **the clarity and quality of the report will be worth 10 pts**. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables. It should be a PDF document.

7. In your report, the **results should always be accompanied by discussions** of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Logistic regression with L2 and L1 regularizations (total points: 90 pts + 10 report pts)

For this assignment, you need to implement and test logistic regression, which learns from a set of N training examples $\{\mathbf{x}_i, y_i\}_{i=1}^N$ an weight vector $\mathbf{w}$ that maximize the log likelihood objective. You will examine two different regularization methods: L2 (ridge) and L1 (Lasso).

Data. This dataset consists of health insurance customer demographics, as well as collected information related to the customers' driving situation. Your goal is to use this data to predict whether or not a customer may be interested in purchasing vehicular insurance as well (this is your "Response" variable). The dataset description (dictionary) is included. **Do not use existing code from outside sources for any portions of this assignment. This would be a violation of the academic integrity policy.**

The data is provided to you in both a training set: **pa2_train.csv**, and a validation set: **pa2_dev.csv**, with an X and y for both (X being features, y being labels). You have labels for both sets of data. The only preprocessing you will need to do is the normalization of the numeric/ordinal features (see next paragraph).

Preprocessing Information In order to train on this data, we have pre-processed it into an appropriate format. This is done for you in this assignment to ensure results are similar across submissions (easier to grade). You should be familiar with this process already from the first assignment. In particular, we have treated **[Gender, Driving_License, Region_Code, Previously_Insured, Vehicle_Age, Vehicle_Damage, Policy_Sales_Channel]** as categorical features. We have converted those with multiple categories (some that originally contained textual descriptions) into one-hot vectors. Note that we left **Age** as an ordinal numeric feature. You are to leave these as is and not modify further for this assignment, but understand the process. **The numeric and ordinal features [Age, Annual_Premium, Vintage] you will need to scale to the range of [0, 1].** Additionally, the dataset should be relatively class balanced (close to the same number of 1's and 0's for Response). This was not the case in the raw data, so we downsampled for easier training purposes. There are other ways to handle class imbalance, beyond the scope of this assignment, but it is a common problem in real-world data.

General guidelines for training. For all parts, you should set a upper limit on the number of training iterations (e.g., 10k) and train your model until either the convergence condition is met, i.e., the improvement of the objective is small, or you hit the iteration limit. If you find that your algorithm needs more than 10k iterations to converge, feel free to use higher values. It is a good practice to monitor objective during the training to ensure that it is not diverging. You will need to adjust your learning rate based on the observed training behavior.

Part 1 (45 pts) : Logistic regression with L2 (Ridge) regularization. Recall, Logistic regression with L2 regularization aims to minimize the following loss function[1]:

$$\frac{1}{N}\sum_{i=1}^{N}\left[-y_i \log \sigma(\mathbf{w}^T\mathbf{x}_i) - (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i)\right] + \lambda\sum_{j=1}^{d} w_j^2 \tag{1}$$

See the following algorithm for batch gradient descent [2] optimization of Equation 1.

Algorithm 1: Gradient descent for Ridge logistic regression

Input: $\{(\mathbf{x}_i, y_i)_{i=1}^N\}$(training data), α(learning rate), λ(regularization parameter)
Output: learned weight vector $\mathbf{w}$
Initialize $\mathbf{w}$;
while *not converged* **do**
  $\mathbf{w} \leftarrow \mathbf{w} + \frac{\alpha}{N}\sum_{i=1}^{N}(y_i - \sigma(\mathbf{w}^T\mathbf{x}_i))\mathbf{x}_i$; // normal gradient without the L2 norm
  for $j = 1$ **to** d **do**
    $w_j \leftarrow w_j - \alpha\lambda w_j$; // L2 norm contribution
  end
end

For this part of the assignment, you will need to do the following:

(a) Implement Algorithm 1 and experiment with different regularization parameters $\lambda \in \{10^{-i} : i \in [0, 5]\}$.

(b) Plot the training accuracy and validation accuracy of the learned model as the λ value varies. What trend do you observe for the training accuracy as we increase λ? Why is this the case? What trend do you observe for the validation accuracy? What is the best λ value based on the validation accuracy?

(c) For the best model selected in (b), sort the features based on $|w_j|$. What are top 5 features that are considered important according to the learned weights? How many features have $w_j = 0$? If we use larger λ value, do you expect more or fewer features to have $w_j = 0$?

[1] In class we presented the log likelihood function as the objective to maximize. It is, however, more common to put a negative in the front and turn it into a loss function, which is called "negative loglikelihood".

[2] Our lecture presented gradient ascent, here since we are working with loss function, we use gradient descent instead.

Part 2 (45 pts). Logistic Regression with L1 (Lasso) regularization For this part, you will need to implement L1 regularized logistic regression. Recall that the loss function for L1 regularized logistic regression is:

$$\frac{1}{N}\sum_{i=1}^{N}\left[-y_i \log \sigma(\mathbf{w}^T\mathbf{x}_i) - (1-y_i)\log(1-\sigma(\mathbf{w}^T\mathbf{x}_i)\right] + \lambda\sum_{j=1}^{d}|w_j| \tag{2}$$

The following algorithm minimizes Equation 2 via a procedure called proximal gradient descent. For L1 regularized loss functions, Proximal gradient descent often leads to substantially faster convergence than simple gradient (or subgradient in this case since the L1 norm is not differentiable everywhere) descent. You can refer to Ryan Tibshirani's note (`http://www.stat.cmu.edu/~ryantibs/convexopt/lectures/prox-grad.pdf`) for an introduction to this method.

Algorithm 2: Proximal gradient descent for LASSO logistic regression

Input: $\{(\mathbf{x}_i, y_i)_{i=1}^N\}$(training data), α (learning rate), λ (regularization parameter)
Output: learned weight vector $\mathbf{w}$
Initialize $\mathbf{w}$;
while *not converged* **do**
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha\frac{1}{N}\sum_{i=1}^{N}(y_i - \sigma(\mathbf{w}^T\mathbf{x}_i))\mathbf{x}_i$; // normal gradient descent without the L1 norm
  for $j = 1$ **to** d **do**
    $w_j \leftarrow sign(w_j)\max\left(|w_j| - \alpha\lambda, 0\right)$; // soft thresholding each w_j: if $|w_j| < \alpha\lambda$, $w_j \leftarrow 0$
  end
end

For this part of the assignment, you will need to do the following:

(a) Implement Algorithm 2 and experiment with different regularization parameters $\lambda \in \{10^{-i} : i \in [0, 5]\}$.

(b) Plot the training accuracy and validation accuracy of the learned model as the λ value varies. What trend do you observe for the training accuracy as we increase λ? Why is this the case? What trend do you observe for the validation accuracy? What is the best value based on the validation accuracy?

(c) For the best model, sort the features based on $|w_j|$. What are top 5 features that are considered important? How many features have $w_j = 0$? If we use larger λ value, do you expect more or fewer features to have $w_j = 0$?

(d) Compare and discuss the differences in your results for Part 1 and Part 2, both in terms of the performance and sparsity of the solution.

Submission. Your submission should include the following:
1) Your source code. **One file for each Part**. The files should be named (for example) **part1.py**, and should run with simply **python part1.py**. You do not need to generate plots in the submission code, please just include those in your report;
2) Your report (see general instruction items 6 and 7 on page 1 of the assignment), which should begin with a general introduction section, followed by one section for each part of the assignment;
3) Please submit the report PDF, along with a .zip containing the code to Canvas. The PDF should be outside the .zip so it's easier to view the report.