CS534 — Implementation Assignment 2 — Due 11:59PM Oct 29st, 2021

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. Submit your report on Canvas and your code on TEACH following this link:
`https://teach.engr.oregonstate.edu/teach.php?type=assignment`.

4. Please follow the submission instructions for file organization (located at the end of the assignment description).

5. Please run your code before submission on the EECS servers (i.e. babylon). You can make your own virtual environment with the packages we've listed in either your user directory or on the scratch directory.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. The report should be clear and concise, with figures and tables clearly labeled with necessary legend and captions. The quality of the report and is worth 10 pts. The report 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: **IA2-train.csv**, and a validation set: **IA2-dev.csv**.

Preprocessing Information We have pre-processed the data into an appropriate format. This is done for you in this assignment to ensure results are similar across submissions (easier to grade). In particular, we have treated [**Gender, Driving_License, Region_Code, Previously_Insured, Vehicle_Age, Vehicle_Damage, Policy_Sales_Channel**] as categorical features and converted those 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. Additionally, the dataset is processed to be relatively class balanced (close to the same number of 1's and 0's for Response). This was not the case in the original raw data, but we downsampled for easier training purposes. Handling class imbalance is beyond the scope of this assignment, but it is a common and important problem in real-world data.

General guidelines for training. For this assignment, you are responsible for finding the right learning rate that works for your 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.

Part 1 (40 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 excluding w_0 for the dummy feature
  end
end

For this part of the assignment, you will need to implement Algorithm 1 and experiment with different regularization parameters $\lambda \in \{10^i : i \in [-3, 3]\}$. This is the minimum range required. Feel free to experiment with more values beyond the specified limits or in-between, especially if it helps you answer some of the questions regarding 'general trends'.

(a) Plot the training accuracy and validation accuracy of the learned model as the λ value varies. You can either plot both in the same figure or in separate figures. If together, please adjust the range for the y-axis to ensure that we can tell them apart. If separate, please align the figures so that we can compare across.

> Question: 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?

(b) Consider the best λ^* selected in (a), a value λ_- that is smaller than λ^*, and a λ_+ that is bigger than λ^*. Report for each of three λ values, the resulting model's top 5 features with the largest weight magnitude $|w_j|$.

> Question: Do you see differences in the selected top features with different λ values? What is your explanation for this behavior?

[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.

(c) For different values of λ, compute the sparsity of the model as the number of weights that equal zero and plot it against λ.

> Question: What trend do you observe for the sparsity of the model as we change λ? If we further increase λ, what do you expect? Why?

Part 2 (40 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 L_1 regularized loss functions, Proximal gradient descent often leads to substantially faster convergence than simple gradient (or subgradient in this case since the L_1 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**
    $w_j \leftarrow sign(w_j)\max(|w_j| - \alpha\lambda, 0)$; // 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 implement Algorithm 2 and experiment with different regularization parameters $\lambda \in \{10^i : i \in [-3, 3]\}$. This is the minimum range required. Feel free to experiment with more values beyond the specified limits or in-between, especially if it helps you answer some of the questions regarding 'general trends'.

(a) Plot the training accuracy and validation accuracy of the learned model as the λ value varies.

> Question: 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?

(b) Consider the best λ^* selected in 2(a), a value λ_- that is smaller than λ^*, and a λ_+ that is bigger than λ^*. Report for each of three λ values, the resulting model's top 5 features with the largest weight magnitude $|w_j|$.

> Question: Do you see differences in the selected top features with different λ values? What is your explanation for this behavior?

(c) For different values of λ, compute the 'sparsity' of the model as the number of weights that equal zero and plot it against λ.

Question: What trend do you observe for the sparsity of the model as we change λ? If we further increase λ, what do you expect? Is this trend different from what you observed in 1(c)? Provide your explanation for your observation.

Part III Kaggle competition. (10 pts) We are hosting a in-class competition using this data. You must participate in single-person teams. The competition link will be posted on canvas. For Part I and II, your are given a small training data and a larger validation data to limit the run-time. For this part, you should decide how to best use the provided labeled data for training and parameter tuning. You can also process the features in different ways, and/or engineer new features. The only limit is that the core algorithm must be your own implementation of the logistic regression algorithm, L1 or L2. For example, you are not allowed to use some more powerful algorithm to train the prediction model.

For this part, you need to describe the methods you used and their performance on the public leaderboard. How do you handle the data usage? Do you treat the features differently from what was used for part I and II? What do you think is limiting the performance you can get on this dataset? The amount of data? the availability of features? the complexity of the algorithm? ...

Competition bonus: The top three performers will receive 2 bonus points.

Submission instructions. Your submission should include the following:
1) Your source codes with a readme file to specify the required packages, zipped in a single file submitted on TEACH. We require **One file for each Part**.
2) You do not need to generate plots in the submission code, please remember to include those in your report. You need to print out your code result of each question in the console since it's easier to check the correctness of your code. Generally speaking, we check the code by console outputs and plots by report.
3) Please do **not** upload your python virtual environment. You can include the **data** in the same directory of the code (unless you think the data file is too big to upload), which can be tested with "python your_code.py" directly.
4) 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; The report should be in PDF, submitted on Canvas.
5) Please always put your name on the first page of the report. All graphs must be properly labeled, i.e. Graphs must contain: Title, labels along both axis. It's a good habit to number pages, sections of the report.