CS534 — Implementation Assignment 2

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 can collaborate in group of up to three people. Please do not share code with other groups, or copy program files/structure from any outside sources like Github. Each group's 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. Your code should follow the structure and function specification of the provided skeleton code.

5. Please test your code before submission on the EECS servers (i.e. babylon) to make sure that it runs correctly on the server and produce the correct results as reported in the your report.

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 a simpler format. 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. 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. There are 3 numerical features, **these need to be normalized so that your training can converge faster.**

General guidelines for training. For this assignment, you are responsible for finding the right learning rate that works for your training. Note that you will need to adjust the learning rate for different λ values. Given that we are experimenting with a very simple learning algorithm that has low complexity. The difference we observe will tend to be quite small. If you don't run your algorithm to close to true convergence, you might not be able to see clear difference in performance.

Part 1 (35 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

[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.

For this part of the assignment, you will need to implement Algorithm 1 and experiment with different regularization parameters $\lambda \in \{10^i : i \in [-4, 2]\}$. This is the minimum range required. Feel free to experiment with more values beyond the specified limits or in-between if it helps you answer the question.

(a) Plot the training accuracy and validation accuracy of the learned model as a function of i used for λ. You can either plot both in the same figure or in separate figures. If separate, please align the figures in your report 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 $|w_j|$ (excluding w_0).

> 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 approximately equal zero ($\leq 10^{-6}$) and plot it as a function of i for 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 (15 pts), Experiment with noisy training data. For this part, you will repeat the experiment of part 1 but using a noisy training set (IA2-train-noisy). This data set is created by flipping the label of 30% of training examples (randomly selected). You will experiment with the same set of λ values as part 1 and plot the training accuracy and validation accuracy of the learned model as a function of i used for λ.

> Question: What are some the key differences do you observe comparing the results obtained using noisy training data to those of part 1? What do you think is the effect of regularization on the model's robustness to noise in the training set? Why?

Part 3 (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 $\lambda \in \{10^i : i \in [-4, 2]\}$ (This is the minimum range required. Feel free to experiment with more values beyond the specified limits or in-between if it helps you answer some of the questions.) on the **clean training data**.

(a) Plot the training accuracy and validation accuracy of the learned model as a function of i used for the λ value.

> 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 $|w_j|$ (excluding w_0).

> 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.

(d) Finally, please compare the results acquired in part 3 with that of part 1.

> What are the key differences between the two regularization methods observed on this data set? Specifically, which method achieves the best validation accuracy? Which method is more sensitive to the choice of the regularization parameter for this data? Which method produced sparser feature weights? What are the advantages and disadvantages of each method in general?