CS689: Machine Learning - Fall 2019

Homework 1

Assigned: Wednesday, Sept 11. Due: Wednesday, Sept 25 at 11:59pm

Getting Started: You should complete the assignment using your own installation of Python 3.6. Download the assignment archive from Moodle and unzip the file. The data files for this assignment are in the `data` Directory. Code templates are in the `code` directory. The only modules allowable during autograding are those already imported in the code templates.

Deliverables: This assignment has two types of deliverables: a report and code files.

- **Report:** The solution report will give your answers to the homework questions (listed below). The maximum length of the report is 5 pages in 11 point font, including all figures and tables. You can use any software to create your report, but your report must be submitted in PDF format. You will upload the PDF of your report to Gradescope under `HW01-Report` for grading. Access to Gradescope will be enabled one week before the assignment is due. For full credit, all figures must have proper axes, labels, legends, and titles. Any tables must have proper row and/or column headings and titles or captions. You do **not** need to include the text of questions in your report.
- **Code:** The second deliverable is your code. Your code must be Python 3.6 compatible (no iPython notebooks, other formats, or code from other versions of Python). You will upload a zip file (not rar, bz2 or other compressed format) containing all of your code to Gradescope under `HW01-Programming` for autograding. Access to the autograder will be enabled one week before the assignment is due. When unzipped, your zip file should produce a directory called `code`. If your zip file has the wrong structure, the autograder may fail to run. To receive credit for implementation questions, your code must run in Gradescope.

Academic Honesty Reminder: Copying solutions from external sources (books, web pages, etc.) or other students is considered cheating. Homework assignments are individual work. Sharing your solutions with other students is considered cheating. Posting your code to public repositories like GitHub is also considered cheating. Collaboration indistinguishable from copying is considered cheating. Any detected cheating will result in a grade of 0 on the assignment for all students involved, and potentially a grade of F in the course.

Questions:

1. (*15 points*) **Optimal Predictions for Zero-One Loss:** Suppose we have a probability distribution $P(Y = y, X = x)$ where $y \in \{0, 1\}$ and $x \in \mathbb{R}$. Suppose we want to find the optimal prediction function $f(x)$ under the expected zero-one prediction loss criteria: $\mathbb{E}_{P(x,y)}[L_{01}(y, f(x))]$. Note that $f : \mathbb{R} \rightarrow \{0, 1\}$. Use this information to answer the following questions.

a. (*5 pts*) Provide an explicit mathematical expression for this optimization objective function in terms of sums and/or integrals, the distribution $P(X = x, Y = y)$ (and/or its marginals or conditionals), the

prediction function $f(x)$, the loss functions L, and the inputs and outputs x and y.

b. (*10 pts*) Prove that the function $f(x)$ that minimizes the expected loss $\mathbb{E}_{P(x,y)}[L_{01}(y, f(x))]$ is given by the expression below. Show your work and explain your solution.

$$f(x) = \arg\max_{y \in \{0,1\}} P(Y = y | X = x)$$

2. (*40 points*) **Logistic Regression with Centering:** In the standard binary logistic regression model, the probability of the class variable $y \in \{-1, 1\}$ is modeled as a logistic function of the input $\mathbf{x}$ and the parameters $\theta = [\mathbf{w}, b]$:

$$P(Y = y | \mathbf{X} = \mathbf{x}, \theta) = \frac{1}{1 + \exp(-y(\mathbf{w}\mathbf{x}^T + b))}$$

A common pre-processing approach when applying machine learning models is to pre-center the inputs so the mean on each dimension is 0. We can instead learn an optimal centering of the inputs by augmenting the logistic regression model with a vector of centering parameters $\mathbf{c}$. The augmented parameter set is $\theta = [\mathbf{w}, \mathbf{c}, b]$ and the augmented model is:

$$P(Y = y | \mathbf{X} = \mathbf{x}, \theta) = \frac{1}{1 + \exp(-y(\mathbf{w}(\mathbf{x} - \mathbf{c})^T + b))}$$

In this question, we will implement and analyze this augmented model. As the learning objective function, we will minimize the regularized negative conditional log likelihood:

$$\mathcal{L}(\mathcal{D}, \theta) = \sum_{n=1}^{N} \log(1 + \exp(-y_n(\mathbf{w}(\mathbf{x}_n - \mathbf{c})^T + b))) + \lambda \|\mathbf{w}\|_2^2 + \lambda \|\mathbf{c}\|_2^2 + \lambda \cdot b^2$$

a. (*5 pts*) Find the gradient of $\mathcal{L}(\mathcal{D}, \theta)$ with respect to $\mathbf{w}$, $\mathbf{c}$, and b. Show your work.

b. (*20 pts*) Starting from the provided template, implement a Scikit-Learn compatible class for this model. For learning, use the fmin_l_bfgs_b optimizer from scipy.optimize with the disp=10 option to see output from the optimizer. Initialize all model parameters to 0. As your answer to this question, describe any issues you had implementing the model and what approach you used to fix them.

c. (*5 pts*) Learn the model on the provided training data set with $\lambda = 1e - 6$, and evaluate it by computing its average prediction error on the test set. The entire learning procedure should take a few seconds. Report the average test error that you find as your answer to this question.

d. (*5 pts*) Using the provided data, learn a standard logistic regression model using scikit-learn's linear_model.LogisticRegression implementation with $C = 10^6$ and solver='lbfgs'. Leave the other model hyper-parameters at their default values. Evaluate the standard model by computing its average prediction error on the test set. Report the average test error that you find as your answer to this question.

e. (*5 pts*)If your implementation is correct, you should find that the augmented and standard models perform nearly identically. Explain why you think this is the case, providing as much detail as possible.

3. (*45 points*) **Regression with Outliers:** One important problem with standard least-squares linear regres-

sion is that outliers can significantly corrupt learning. One way to combat this problem is with regression loss functions that are less sensitive to larger errors. Consider the robust prediction loss function shown below where $\delta > 0$ is a parameter of the loss.

$$L_\delta(y, y') = \delta^2(\sqrt{1 + (y - y')^2/\delta^2} - 1)$$

Using this robust loss for a linear regression model $f(\mathbf{x}_n, \theta) = \mathbf{w}\mathbf{x}_n^T + b$ results in the learning objective

$$\mathcal{L}_\delta(\mathcal{D}, \theta) = \sum_{n=1}^{N} L_\delta(y_n, f(\mathbf{x}_n, \theta))$$

Use this model, objective function, and loss to answer the following questions. Assume that $\mathbf{x} \in \mathbb{R}^D$ and $y \in \mathbb{R}$ in your derivations and code.

a. (*5 pts*) Produce a plot of the robust loss function $L_\delta(y, y')$ for $\delta \in \{0.1, 1, 10\}$ where the x-axis corresponds to the prediction error $\epsilon = y - y'$ and the y-axis corresponds to the value of the loss. In the same figure, also include a similar plot of the standard squared loss $L_{sqr}(y, y') = (y - y')^2$. Use these plots to explain why the loss $L_\delta(y, y')$ could result in less sensitivity to outliers than the squared loss $L_{sqr}(y, y')$.

b. (*5 pts*) Find the gradient of $\mathcal{L}_\delta(\mathcal{D}, \theta)$ with respect to $\mathbf{w}$ and b. Show your work.

c. (*20 pts*) Starting from the provided template, implement a Scikit-Learn compatible class for this model. For learning, use the fmin_l_bfgs_b optimizer from scipy.optimize with the disp=10 option to see output from the optimizer. Initialize all model parameters to 0. As your answer to this question, describe any issues you had implementing the model and what approach you used to fix them.

d. (*5 pts*) Using the provided data set, apply the model using $\delta = 1$. Also apply Scikit-Learn's standard least-squares linear regression model linear_model.LinearRegression. Report the MSE achieved by both models on the train data set.

e. (*5 pts*) Provide a single figure that includes a scatter plot of the training data, the regression line found using the robust model (with $\delta = 1$, and the regression line found using the standard least-squares model.

f. (*5 pts*) Based on this plot, which model do you think would have better generalization performance on future test data? Explain your answer.