CS689: Machine Learning - Fall 2018

Homework 1

Assigned: Wednesday, Sept 12. Due: Wednesday, Sept 26 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.

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.
- **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.

Academic Honesty Statement: Copying solutions from external sources (books, web pages, etc.) or other students is considered cheating. 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. (*20 points*) **An Alternative Bernoulli Model:** Consider the probability mass function for $X \in \{0, 1\}$ shown below, which is an alternate parameterization of the standard Bernoulli distribution. Use this model to answer the following questions.

$$P(X = x) = \left(\frac{1}{1 + \exp(-\lambda)}\right)^{[x=1]} \left(\frac{1}{1 + \exp(\lambda)}\right)^{[x=0]}$$

a. (*5 pts*) Show that this distribution is properly normalized.

b. (*10 pts*) Suppose we have a data set $\mathcal{D}$ containing a one's and b zero's. Derive the MLE of λ given $\mathcal{D}$ as

the data set. You may assume that a and b are both greater than zero. Show your work.

c. (*5 pts*)
What is the mathematical relationship between this parameterization of the Bernoulli distribution and the standard parameterization of the Bernoulli distribution? In particular, give an expression for λ in terms of θ and an expression for θ in terms of λ. Briefly explain your answer.

2. (*40 points*) **Logistic Regression with an Informative Prior:** Recall that standard ℓ_2 regularization for logistic regression is equivalent to MAP estimation of the model parameters under a zero-mean, spherical Gaussian prior. In some situations, including transfer learning, we may have an informative prior for the model parameters including a non-zero prior mean $[\mathbf{w}_0, b_0]$. Assuming a spherical Gaussian around this mean results in the following learning objective function. Use this learning objective to answer the following questions.

$$\mathcal{L}(\mathcal{D}, \theta) = \sum_{n=1}^{N} \log(1 + \exp(-y_n(\mathbf{w}\mathbf{x}_n^T + b))) + \lambda\|\mathbf{w} - \mathbf{w}_0\|_2^2 + \lambda(b - b_0)^2$$

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

b. (*20 pts*) Starting from the provided template, implement a Scikit-Learn compatible class for this model including fit, predict, set_params, and get_params, as well as functions for computing the objective function and gradient. As your answer to this question, describe your approach to learning the model parameters in your report and submit your commented code for auto grading as described above.

c. (*10 pts*) Using the provided data and prior mean parameters $\mathbf{w}_0$, b_0, perform an experiment where the model is learned using $\lambda = 0$ and $\lambda = 10$ while varying the number of training cases from 10 to 400 in steps of 10. Provide one figure containing two line plots showing the test accuracy as a function of the number of training data cases for both values of λ.

d. (*5 pts*) Explain why the relative performance of using $\lambda = 0$ and $\lambda = 10$ changes as the amount of training data increases.

3. (*40 points*) **Generalized Robust Regression:** One important problem with standard least-squares linear regression is that outliers can significantly corrupt learning. Consider the generalized robust regression loss function shown below. The loss switches from being a polynomial of degree $2k$ to a linear function at the point δ. Both the point δ and the polynomial order parameter k are parameters of the this loss. δ is a non-negative real number and k is a positive integer ($k = 1, 2, 3...$).

$$L_{k,\delta}(y, y') = \begin{cases} \frac{1}{2k}(y - y')^{2k} & \text{... if } |y - y'| \leq \delta \\ \delta^{2k-1}(|y - y'| - \frac{(2k-1)}{2k}\delta) & \text{... otherwise} \end{cases}$$

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

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

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

b. (*20 pts***)** Starting from the provided template, implement a Scikit-Learn compatible class for this model including fit, predict, set_params, and get_params, as well as functions for computing the objective and gradient. As your answer to this question, describe your approach to learning the model parameters in your report and submit your commented code for auto grading as described above.

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

d. (*5 pts***)** Provide a single figure that includes a scatter plot of the training data, the regression line found using the robust model, and the regression line found using the standard least-squares model.

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