EE 516: Mathematical Foundations of Machine Learning **Winter 2023**

Homework 8
Due: March 12, 2023, 11:59PM PT

Student Name: *Instructor Name: John Lipor*

Problem 1 (5 pts, 2 pts)

In Homework 6, problem 3, you implemented a majorize-minimize algorithm for solving the mean absolute error minimization problem. The resulting algorithm fit the iteratively reweighted least squares (IRLS) formulation. In this problem, you will implement the IRLS algorithm directly for minimizing the p-norm of the residuals, i.e., you will attempt to solve the problem

$$\min_{x \in \mathbb{R}^D} \|Ax - b\|_p^p .$$

(a) Implement the IRLS algorithm for ℓ_p regression described in Algorithm 1 of the IRLS notes by completing the `lpIRLS` function. Note the stopping criterion (for yourself). **Turn in** your code.

(b) Test your algorithm by running `lpIRLS` for $p \in \{0.1, 0.5, 0.7, 1\}$ in `prob1`. **Turn in** the resulting plot and state which value of p performed best. Why do you think this is the case?

Problem 2 (5 pts, 3 pts)

In this problem, you will implement an IRLS algorithm for solving sparse regression problems. Recall that in the IRLS lecture notes we developed a method of (approximately) solving

$$\begin{aligned} \min_{x \in \mathbb{R}^D} \quad & \|x\|_p \\ \text{subject to} \quad & Ax = b, \end{aligned}$$

where we typically consider highly under-determined systems so that the equality constraint can be satisfied. When $p = 1$, this optimization problem is referred to as *basis pursuit*.

(a) Implement the IRLS algorithm for sparse regression described in Algorithm 2 of the IRLS notes by completing the `bpIRLS` function. Note the stopping criterion (for yourself). **Turn in** your code.

(b) One important way of visualizing algorithm performance is through the use of a *heatmap*, which shows the error as a function of two algorithm parameters in the form of an image. The error of interest in this case is

$$\frac{\|x_{\text{true}} - \hat{x}\|_2}{\|x_{\text{true}}\|_2},$$

where $\hat{x}$ is the output of your `bpIRLS` algorithm. The parameters of interest are the *sparsity* (number of nonzero elements in x_{true}) and the *number of measurements* N (where $A \in \mathbb{R}^{N \times D}$). To test your sparse regression algorithm, display a heatmap by completing the `srHeatmap` script. Let the horizontal axis be sparsity/D and the vertical axis be N/D. **Turn in** heatmaps for $p \in \{0.1, 1, 2\}$ and a description of when the recovery error is low (in terms of sparsity and N) for each value of p.
Note that it is common to divide both sparsity and number of measurements by the problem size D, which is a form of normalization. If time permits (optional), change the value of D and inspect the resulting plots. You will notice that they look exactly the same.

Problem 3 (5 pts, 3 pts)

In this problem, you will implement an ADMM algorithm for solving the Lasso, which is an alternative formulation of the sparse regression problem. Recall that in the ADMM lecture notes we developed a method of solving

$$\min_{x \in \mathbb{R}^D} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 .$$

Using the *variable splitting* technique, we derived the update equations for ADMM to solve this problem.

(a) Implement the ADMM algorithm for solving the Lasso described in the ADMM notes by completing the `lassoADMM` function. Note the stopping criterion (for yourself). **Turn in** your code.

(b) An important part of using algorithms like ADMM is choosing the tuning parameters. For ADMM, it is common to choose ρ as a function of λ as I have done in the given file. Use the script `srHeatmap` to try several values of λ and note any trends that you see. **Turn in:**

- the heatmap for the best value of λ that you found with your choice of λ specified in either the title or the figure caption
- a sentence or two describing how the results change when λ is both smaller and larger than your selected value
- a sentence comparing the performance of ADMM and IRLS for sparse regression.

Problem 4 (10 pts, 5 pts, 2 pts)

As mentioned in class, sparse regression can be used as a tool for feature selection in machine learning. The idea is to start with a large number of features, some of which may not be useful for prediction, and automatically determine which are most valuable for predicting the given labels. Follow the `prob4` script to load the Residential Building dataset from the UCI Machine Learning Repository. This dataset includes a variety of features that can be used to predict the sale price of the corresponding building. You can open the xlsx file to view the actual features. You will use your `lassoADMM` function to perform feature selection on this dataset.

(a) Split the data into training, validation, and test sets. Use the first 250 examples for training, the next 50 for validation, and the last 72 for testing. Run your `lassoADMM` script for 20 values of λ spaced linearly between 1,000-10,000 and plot the squared error of your learned regression function $\hat{w}$ on both the training and validation sets. The validation error can be used to select the "best" choice for λ in terms of generalizing to unseen data. **Turn in** your plots, the best λ value, and a comment on the what you notice about the training and validation error plots.

(b) Now create a stem plot of your learned regressor $\hat{w}$ for the choice of λ you found above. Manually examine the features corresponding to the elements of $\hat{w}$ with largest magnitude and list the top 5 here.

(c) Report the error on the test dataset for your chosen value of λ.

Problem 5 (5 pts)

Read the article, "Who Should Stop Unethical A.I.?" linked here and also available on the `interesting-reading` channel in pdf format (in case you're paywalled). After reading, share one major ethical issue/concern you learned about from the article and one idea (whether your own or from the article) for mitigating unethical A.I. in the `interesting-reading` channel on Slack.