15.095: Machine Learning Under a Modern Optimization Lens **Fall 2021**

Homework 1: Due September 22

Hand in: **pdf** upload to Canvas. Please append any Julia code **at the end** of the whole pdf.

Note: this homework covers the first three lectures and two recitations. We recommend attempting questions after the relevant content has been covered in class.

1.1 Question 1: True/False (30 marks)

Please classify the following statements as true or false and justify your answer. If the statement is false, please provide a counter example. We will assign 2 marks for correctly classifying the answer, and 3 marks for the validity of the justification/counterexample.

(a) The following optimization problem can be reformulated exactly as a linear optimization problem:

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & 2x_1 + 3|x_2 - 10| \\ \text{s.t.} \quad & |x_1 + 2| + |x_2| \leq 5. \end{aligned}$$

(b) The following optimization problem can be reformulated exactly as a linear optimization problem:

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & 2x_1 + 3|x_2 - 10| \\ \text{s.t.} \quad & |x_1 + 2| + |x_2| \geq 5. \end{aligned}$$

(c) If a primal linear optimization problem is unbounded, then its dual problem is infeasible.

(d) The main advantage of Lasso over ordinary least squares regression (without regularization) is that it provides sparser solutions.

(e) Stable linear regression reduces to a linear optimization problem.

(f) Suppose that we are given m constraints $\boldsymbol{a}_i^\top \boldsymbol{x} \geq b_i,\ i \in \{1, \ldots, , m\}$ where $\boldsymbol{a}_i \in \mathbb{R}^n$. Moreover, suppose that there exists some finite M such that $\boldsymbol{a}_i^\top \boldsymbol{x} \geq M$ for any feasible $\boldsymbol{x}$. Then, the requirement that at least k of these constraints are satisfied can be modeled via linear integer optimization.

(g) The following function is convex on $\{(\boldsymbol{x}, \boldsymbol{X}) \in \mathbb{R}^n_+ \times \mathcal{S}^n_{++}\}$, where $\mathcal{S}^n_{++}$ denotes the set of positive definite matrices:

$$f(\boldsymbol{x}, \boldsymbol{X}) = \boldsymbol{x}^\top \boldsymbol{X}^{-1} \boldsymbol{x}$$

[Note: part (g) is a bonus question, you can earn up to an additional five marks by answering it, provided your total grade does not exceed 100. However, it is also harder than the rest of the pset.]

1.2 Question 2: Linear Regression (40 marks)

Assume that we are given a data set for regression $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$, and we would like to predict y given $\mathbf{x}$. The Ordinary Least Squares (OLS) estimator is:

$$\hat{\beta}^{LS} \in arg\,\underset{\beta}{min} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \beta\right)^2$$

where $\left(y_i - \mathbf{x}_i^T \beta\right)^2, i = 1, \ldots, n$, are the squared residuals.

In this question, we study linear regression variants that can be formulated as linear optimization problems.

(a) (10 marks) Formulate the problem of minimizing the maximum of the absolute residuals $\left|y_i - \mathbf{x}_i^T \beta\right|$, i.e.,

$$\min_{\boldsymbol{\beta}} \max_{i \in \{1,\ldots,n\}} \quad \left|y_i - \mathbf{x}_i^T \beta\right|,$$

as a linear optimization problem.

(b) (10 marks) The previous formulation is very sensitive to noise. Using your answer to part (a), design a new optimization formulation which is more robust yet comparably cheap to optimize, i.e., is still a linear optimization problem with a comparable number of variables.

(c) (10 marks) Propose an alternative robust optimization formulation which has both linear and quadratic terms in the objective. What is one advantage and one disadvantage of this formulation over the formulation in (b).

(d) (10 marks) After a quick check of the correlation matrix, you realize that features 15 and 95 are highly correlated. How can you modify the previous formulation so that you only use at most one of them?

1.3 Question 3: Stable Regression (30 marks)

(a) (5 marks) Using `Julia/JuMP`, write a function to solve the following problem:

$$\min_{\beta} \sum_{i=1}^{n} |y_i - \beta_0 - \boldsymbol{\beta}^\top \boldsymbol{x}_i| + \lambda \sum_{i=0}^{p} |\beta_i|, \tag{1.1}$$

(b) (5 marks) Using this function, fit the linear regression model in equation 3 on the data in the files `stableX_train_and_valid.csv`, `stableY_train_and_valid.csv`, `stableX_test.csv` and `stableY_test.csv` (which were created using the Abalone dataset from the UCI Machine Learning Repository [1]).

- For the data from the `train_and_valid` files, randomly split it into 70% training and 30% validation.
- Use the validation set to select an optimal value for the parameter λ in the set $\{0.01, 0.03, 0.08, 0.1, 0.3, 0.8, 1, 3\}$.
- Try recombining the training and validation sets and then splitting them up in a different manner a few times (at least 5), and then training/validating a new model.
- What is the range of Mean Squared Errors (MSEs) you get on the test data for each of these models?

(c) (5 marks) Given the robust optimization problem

$$\min_{\beta} \max_{z \in \mathcal{Z}} \sum_{i=1}^{n} z_i |y_i - \beta_0 - \boldsymbol{\beta}^\top \boldsymbol{x}_i| + \lambda \sum_{i=0}^{p} |\beta_i|,$$

where

$$\mathcal{Z} = \left\{ z : \sum_{i=1}^{n} z_i = k, z_i \in \{0, 1\} \right\},$$

write down an equivalent linear optimization problem. Please write detailed explanations.

(d) (15 marks) Using `Julia/JuMP`, solve the optimization problem from part (c), setting k to be the number of points in the training set, and using λ's in the set $\{0.01, 0.03, 0.08, 0.1, 0.3, 0.8, 1, 3\}$.

- For each λ, use the solution to divide the data into training and validation sets, and calculate the MSE of the model on the validation set.
- Choose the λ corresponding to the lowest MSE as your optimal λ.
- Then, use the function from part (a) on the combined training/validation set with the optimal λ you found to create your final model.
- What is the MSE of this new model on the test data?
- Compare it to your results from part (b) – how do the MSEs compare?

References

[1] D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.