Convex Optimization Homework Assignment #1

Due: Friday April 25, 2025, 6am.

1. (50%) In this assignment, you will analyze the convexity of a regularized logistic regression problem and implement its solution using Python and CVXPY. The tasks include:
   - Analyzing the convexity of the objective function.
   - Writing and running Python code using CVXPY to solve the problem.
   - Understanding the *Disciplined Convex Programming (DCP)* rules in CVXPY.

Problem Description

You are tasked with fitting a logistic regression model with **L2 regularization** to a dataset with m data points:

- $x_i \in \mathbb{R}^n$ is the feature vector for the i-th data point.
- $y_i \in \{0, 1\}$ is the label associated with x_i.

The objective is to minimize the regularized logistic loss:

$$\text{minimize } f(\beta) = \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp(-y_i x_i^T \beta)\right) + \lambda \|\beta\|_2^2 \tag{1}$$

where:

- $\beta \in \mathbb{R}^n$ is the vector of model coefficients.
- $\lambda > 0$ is the regularization parameter controlling the strength of the penalty.

(a) (10%) Prove that the logistic loss function

$$f_i(\beta) = \log\left(1 + \exp(-y_i x_i^T \beta)\right) \tag{2}$$

is convex. *Hints:*

- *The exponential function is convex and non-decreasing.*
- *The composition of a convex function with an affine function remains convex.*

(b) (5%) Prove that $f(\beta)$ as in Eq. (1) is convex.

(c) (35%) Use CVXPY to solve Problem (1) with the following Data set using CVXPY.

Note: CVXPY uses **Disciplined Convex Programming (DCP)** rules to ensure that a problem is correctly formulated. To verify DCP compliance, ensure the following:

- The objective is a convex function to be minimized.
- The constraints (if any) are convex functions with valid affine or inequality forms.
- All operations follow DCP-compliant compositions.

Data Set 1

Please use $n = 5, m = 10$, x_i being the ith column of X matrix

$$X = \begin{bmatrix} -2.05285132 & -0.0798365 & 0.88811551 & -0.31296891 & 0.10815235 \\ -0.31591073 & -0.1387073 & -1.30701794 & -0.63227706 & -0.66792228 \\ 0.3301553 & -0.95748847 & 0.84747887 & -0.58438193 & -0.24445801 \\ -0.38815689 & -1.31530838 & 1.17174758 & 0.43759445 & 1.1907391 \\ 0.67756632 & -0.29914331 & -0.5395671 & 0.20478403 & -0.87697091 \\ -0.05352987 & -2.07222594 & -0.41261809 & 0.43838812 & -0.67987968 \\ -0.56141173 & 0.71125867 & -0.97146703 & 0.32358934 & 0.14434927 \\ -0.09415079 & 0.42187351 & -0.43039491 & 0.6580349 & -0.18454576 \\ 0.30693953 & 1.17048685 & 1.69235586 & 0.56215282 & 0.22329091 \\ -1.32795384 & -1.15396708 & 1.93912882 & -0.67644897 & 0.5756599 \end{bmatrix}^T,$$

and y_i being the ith component of $y = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T$. Write down the optimal value $f(\beta)$ and optimal point β you obtained.

(d) (20%) Repeat (c) with a larger data set specified below.

Data Set 2

Please use $n = 50, m = 100$, and let x_i and y_i be randomly generated (using, e.g., `randint` and `randn` functions from `numpy.random`). Please record the matrix and vector X, y you generated, along with the optimal β^* as three seprate csv files.

Hint: You can use the following command to save a numpy matrix or vector as a csv:

```
np.savetxt("X.csv", X, delimiter=",", fmt="%.5f")
np.savetxt("y.csv", y, delimiter=",", fmt="%.5f")
np.savetxt("beta.csv", beta, delimiter=",", fmt="%.5f")
```

Deliverables

- Submit a PDF report containing:
  - Your convexity analysis and justification. (i.e., Problems (a)(b))
  - The answer you obtained for Problem (c).
- Submit a *.py file for each of Problems (c) and (d) (named `P1_c.py` and `P1_d.py` respectively)
- Submit the three *.csv files for Problem (d).

2. (50%) In this exercise, you will explore a convex optimization problem whose optimal set contains multiple (in fact, infinitely many) solutions. You will:

- Solve the problem using CVXPY and interpret the result.
- Determine whether the solution returned by the solver is unique.
- Characterize the full set of optimal solutions.
- Visualize the feasible region and optimal set.

Problem Statement

Let $x = [x_1, x_2]^T \in \mathbb{R}^2$. Solve the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & (x_1 + x_2 - 1)^2 \\ \text{subject to} \quad & x_1 + x_2 = 2 \\ & x_1 \geq 1 \\ & x_2 \geq 0 \end{aligned}$$

(a) (10%) **Solve Using CVXPY**: Write Python code using CVXPY to solve the problem and report the optimal value and optimal solution.

(b) (10%) **Analyze Uniqueness**:
Answer the following:
- (5%) Is the optimal solution unique? Justify your answer.
- (5%) Why might the solver only return one solution despite the existence of multiple optimal points?

(c) (10%) **Characterize the Optimal Set**
- (5%) Describe the full set of optimal points mathematically.
- (5%) Provide a parametric expression or geometric description of the optimal set.

(d) (10%) Plot the feasible set and highlight the optimal set. You may plot by hand, and you may also use matplotlib or another plotting library.

(e) (10%) **Perturbation**: Modify the objective function slightly to be $(x_1 + x_2 - 1)^2 + \epsilon x_2^2$ with a small $\epsilon > 0$. Solve again and describe how the solution changes. This helps reveal the idea of *regularization* or *tie-breaking* in solvers.
- (5%) Is the optimal solution unique? Justify your answer.
- (5%) Let $\epsilon = 1, 10^{-4}$ and 10^{-10}, respectively. What answers of optimal points and optimal set did you observe?

Deliverables

- Submit two *.py files for Problems (a)(e) (named `P2_a.py` and `P2_e.py` respectively).
- Submit a *.pdf file for Problems (b)(c)(d)(e).

Homework submission guidelines:

- Submit your answer online as a set of eight files: four *.py files, three *.csv files, and a document file (in *.pdf) that contains all answers in this problem set.
- Submit your files online at the NTU Cool website. No paper shall be handed in.
- Late submissions will be treated under the principle elaborated as follows.
  (1) Homework received by 11pm, April 24 (t_1) will be counted fully.
  (2) Homework received after 2am, April 25 (t_2) will not be counted.
  (3) Homework received between t_1 and t_2 will be counted with a discount rate
  $$\frac{t_2 - t}{t_2 - t_1}$$
  where t is the received time. Note that $t_2 - t_1$ is three hours.
- Plagiarism is strongly prohibited. While discussions among classmates are allowed (and encouraged), you shall not ask anyone else to share his/her codes with you, **nor should you attempt to share with anyone your codes (in other words, it is your DUTY to ALWAYS REFUSE TO ANY CLASSMATE WHO ASKS YOU TO SHARE YOUR CODE WITH HIM/HER)**. You should write every line of the code by yourself. If any part of your submission is found to be copied from someone else's submission, then both of your homework submissions will be counted zero.