ORF 523	Problem set 3	Spring 2024, Princeton University
Instructor: A.A. Ahmadi		
AIs: A.Z. Chaudhry, Y. Hua		
Due on March 21, 2024, at 1:30pm EST, on Gradescope		

Problem 1: Norms defined by convex sets

Define M_C as the following function of a convex set C in $\mathbb{R}^n$:

$$M_C(x) = \inf\{t > 0 \mid \ \frac{x}{t} \in C\},$$

over the domain

$$dom(M_C) = \{x \in \mathbb{R}^n \mid \frac{x}{t} \in C \text{ for some } t > 0\}.$$

1. Show that M_C is a convex function.
2. Suppose C is also compact, origin symmetric ($x \in C$ if and only if $-x \in C$), and has nonempty interior. Show that M_C is a norm over $\mathbb{R}^n$. What is its unit ball?
3. Show that an even degree homogeneous polynomial is convex if and only if it is quasiconvex. (Hint: use what you proved in the previous parts of this question.)

Problem 2: Totally unimodular matrices

Let A be an integral matrix. Show that A is totally unimodular if and only if for every integral vector b, the polyhedron $\{x \mid x \geq 0, Ax \leq b\}$ is integral. (Hint: If A is not totally unimodular, use the inverse of a submatrix which does not have determinant $\{0, -1, +1\}$ to construct an integer vector b that generates a non-integral vertex in the polyhedron.)

Problem 3: Radiation treatment planning (Boyd&Vandenberghe, Additional Exercises, Problem 17.2)

In radiation treatment, radiation is delivered to a patient, with the goal of killing or damaging the cells in a tumor, while carrying out minimal damage to other tissue. The radiation is delivered in beams, each of which has a known pattern; the level of each beam can be adjusted. (In most cases multiple beams are delivered at the same time, in one 'shot', with the treatment organized as a sequence of 'shots'.) We let b_j denote the level of beam j, for $j = 1, \dots, n$. These must satisfy $0 \leq b_j \leq B^{\max}$, where $B^{\max}$ is the maximum possible beam level. The exposure area is divided into m voxels, labeled $i = 1, \dots, m$. The dose d_i delivered to voxel i is linear in the beam levels, i.e., $d_i = \sum_{j=1}^{n} A_{ij} b_j$. Here $A \in \mathbb{R}_+^{m \times n}$ is a (known) matrix that characterizes the beam patterns. We now describe a simple radiation treatment planning problem.

A (known) subset of the voxels, $\mathcal{T} \subset \{1, \dots, m\}$, corresponds to the tumor or target region. We require that a minimum radiation dose D^{target} be administered to each tumor voxel, i.e., $d_i \geq D^{\text{target}}$ for $i \in \mathcal{T}$. For all other voxels, we would like to have $d_i \leq D^{\text{other}}$, where D^{other} is a desired maximum dose for non-target voxels. This is generally not feasible, so instead we settle for minimizing the penalty

$$E = \sum_{i \notin \mathcal{T}} (d_i - D^{\text{other}})_+,$$

where $(\cdot)_+$ denotes the nonnegative part of its argument (i.e., $(z)_+ = \max\{0, z\}$). We can interpret E as the total nontarget excess dose.

1. Show that the treatment planning problem is a linear program. The optimization variable is $b \in \mathbb{R}^n$; the problem data are $B^{\max}$, A, $\mathcal{T}$, D^{target}, and D^{other}.

2. Solve the problem instance with data generated by the file `treatment_planning_data.m`. If you are using Python, you can download `Atumor.csv` and `Aother.csv`, then use `treatment_planning_data.py` to load them. Here we have split the matrix A into `Atarget`, which contains the rows corresponding to the target voxels, and `Aother`, which contains the rows corresponding to other voxels. Plot the dose histogram for the target voxels, and also for the other voxels. (You can use the MATLAB function `hist` or the Python function `matplotlib.pyplot.hist` to plot histograms.) Make a brief comment on what you see.

   Remark: The beam pattern matrix in this problem instance is randomly generated, but similar results would be obtained with realistic data.

Problem 4: Support Vector Machines (SVMs)

Recall our Support Vector Machines application of convex optimization from lecture. We have m feature vectors $x_1, \ldots, x_m \in \mathbb{R}^n$ with each x_i having a label $y_i \in \{-1, 1\}$. The goal is to find a linear classifier, that is a hyperplane $a^T x - b$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, by solving the optimization problem[1]

$$\begin{aligned} \min_{a,b}.\ & \|a\| \\ \text{s.t. } & y_i(a^T x_i - b) \geq 1 \text{ for } i = 1, \ldots, m. \end{aligned} \tag{1}$$

We will then use this classifier to classify new data points.

1. Prove that the solution to (1) is unique.

2. We would like to show that the optimization problem (1) is equivalent to

$$\begin{aligned} \max_{a,b,t}.\ & t \\ \text{s.t. } & y_i(a^T x_i - b) \geq t \text{ for } i = 1, \ldots, m \\ & \|a\| \leq 1, \end{aligned} \tag{2}$$

   which is easier to interpret in terms of finding a classifier with maximum margin. Show that if (1) is feasible (with a positive optimal value), then (2) is feasible (and has a positive optimal value). Conversely, show that if (2) is feasible (with a positive optimal value), then (1) is feasible (and has a positive optimal value).

3. Assume the optimal value of (2) is positive. Show that an optimal solution of (2) always satisfies $\|a\| = 1$.

4. Prove that the Euclidean distance of a point $v \in \mathbb{R}^n$ to a hyperplane $a^T z = b$ is given by

$$\frac{|a^T v - b|}{\|a\|}.$$

[1] You can assume throughout that there is at least one data point with $y_i = 1$ and one with $y_i = -1$ as otherwise there is nothing to classify.

Problem 5: Hillary or Bernie?

You would like to use the knowledge you've acquired in optimization over the past few weeks to see if you could have predicted the outcome of each Hillary-Bernie race in the Democratic primaries. To make things easier, you consider only the counties in the tri-state area and New England, i.e., those that belong to the states of New York, New Jersey, Maine, New Hampshire, Pennsylvania, Vermont, Massachussetts, Connecticut, or Rhode Island.

Your goal is to find a linear classifier that, for each county, labels it either as a Bernie win or as a Hillary win. To do this, you have access to a feature vector comprising the following features: mean income, percentage of hispanics, percentage of whites, percentage of residents with a Bachelor's degree or higher, and population density.

1. Load the data file `Hillary_vs_Bernie` in MATLAB. For Python users, you can use the following code to load the data file.

```
import scipy
mat = scipy.io.loadmat('Hillary_vs_Bernie.mat')
X = mat['features_train']
y = mat['label_train']
```

   In `features_train`, we have given you the feature vectors for 175 counties and in `label_train`, their corresponding labels (-1 is a Bernie win and 1 is a Hillary win). As there was a wide disparity in the orders of magnitude of the original data, each feature vector has already been normalized by its standard deviation. The original data can be found at `https://www.kaggle.com/benhamner/2016-us-election`. Solve the problem below:

$$
\begin{aligned}
&\min_{a,b,\eta}. \|a\| + \gamma\|\eta\|_1 \\
\text{s.t. } & y_i(a^T x_i - b) \geq 1 - \eta_i \text{ for all } i = 1, \dots, m \\
& \eta_i \geq 0 \text{ for all } i = 1, \dots, m.
\end{aligned}
$$

   to build a linear classifier for this training set for $\gamma = 0.1, 1, 10$. For each value of γ, specify the optimal a^* and b^* obtained.

2. Test the performance of your classifier using the feature vectors from 21 other counties (given in `features_test`) by comparing the labels obtained to the ones given in `label_test`. Which γ gives you the highest success rate in terms of prediction? Take a look at the entries of a^* in this case – what does this suggest about the people who vote for Hillary compared to those who vote for Bernie?