Convex sets, functions and problems

1. Consider the following optimization problem for which the decision variable is $[x_1, x_2]^T$.

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & f_1(x) = \dfrac{x_1}{1 + x_2^2} \le 0 \\ & h_1(x) = (x_1 + x_2)^2 = 0. \end{array}$$

   (a) Explain why this is **not** a convex problem.

   (b) Determine (by hand) what the feasible set of this program is, i.e. the set $\left\{x \in \mathbb{R}^2 \mid f_1(x) \le 0, \quad h_1(x) = 0\right\}$ and use it to write down an equivalent but convex problem.

2. **Hyperbolic constraints and second-order cones** [B&V, 4.26]

   (a) Show that $x \in \mathbb{R}^n$ and and $y \in \mathbb{R}$, $z \in \mathbb{R}$ satisfy

$$x^T x \le yz, \quad y \ge 0, z \ge 0$$

   if and only if

$$\left\| \begin{bmatrix} 2x \\ y - z \end{bmatrix} \right\|_2 \le y + z,$$

   Use this to show that the following problems can be rewritten as second-order cone programs:

   i. *Maximizing the harmonic mean*

$$\text{maximize} \quad \left(\sum_{i=1}^{m} \frac{1}{(a_i^T x - b_i)} \right)^{-1}$$

   with domain $\{x \mid Ax > b\}$ where a_i^T is the i^{th} row of A.

   ii. *Maximizing the geometric mean*

$$\text{maximize} \quad \left(\prod_{i=1}^{m} (a_i^T x - b_i) \right)^{1/m}$$

   with domain $\{x \mid Ax > b\}$ where a_i^T is the i^{th} row of A.

3. **Support functions**

   The support function of a set C is defined as

$$S_C(y) = \sup \{ y^x \mid x \in C \}.$$

   We allow the function S_C to take the value ∞ where appropriate. Show the following:

   (a) The function S_C is convex, regardless of the convexity of C.

   (b) $S_C = S_{\text{conv}(C)}$, where $\text{conv}(C)$ is the convex hull of C.

4. ***Largest-L* norm of a vector**

For a vector $x \in \mathbb{R}^n$, suppose that $x_{[i]}$ is the i^{th} largest element in magnitude, i.e.

$$|x_{[1]}| \geq |x_{[2]}| \geq \cdots \geq |x_{[n]}| \geq 0.$$

The largest-L norm of x is defined as

$$\|x\|_{[L]} = \sum_{i=1}^{L} |x_{[i]}|$$

(a) Show that $\|x\|_{[L]}$ is a convex function.

(b) Show that $\|x\|_{[L]}$ can be computed by solving an integer programming problem.

(c) Show that $\|x\|_{[L]}$ can be computed by solving a linear program.
(Hint : you can use part (ii) of the support function result above).

Applications

1. Formulate and solve the Sudoku problem in the course notes as an integer program.
2. In this problem you will use a support vector machine (SVM) method to classify images from the MNIST data set as discussed in the course lectures.
   (a) Obtain the MNIST data set from

       `https://www.kaggle.com/datasets/oddrationale/mnist-in-csv`

       in CSV format.
   (b) For a single value (say 5, but it doesn't matter which you pick), load the examples from the MNIST training and test sets that correspond to images of the value 5. Note that the first entry in every row of the MNIST data set gives the value in the image, while the rest of the row is a vector with 284 (i.e. 28×28) entries representing the image of the number as a column vector.
   (c) Use the *training data* set to train a SVM model to classify each image in the training set as "5" or "not 5". You will need to do this parametrically for increasing values of λ. Choose the value λ that gives the best classification rate for the *test data* set.
   (d) Do the same thing for all of the other numbers 0–9.
   (e) Apply the SVM models you have computed for the complete set of values 0–9 to the full test data set. For each entry in the test set, classify it as a number 0–9. What is your overall success rate?

Duality

1. **Projection onto the ℓ_1 ball** [B&V extras, 5.5]

   Consider the problem of projecting a point $a \in \mathbb{R}^n$ onto the unit ball in the ℓ_1 norm:

   $$\begin{array}{ll} \underset{x}{\text{minimize}} & \frac{1}{2}\|x-a\|_2 \\ \text{subject to} & \|x\|_1 \leq 1 \end{array}$$

   - Derive the dual problem and describe an efficient method for solving it.
   - Explain how to find the optimal x from the solution of the dual problem.

2. **SVM duality**

   In the course lectures we saw that a standard support vector machine problem can be written as

   $$\min_{a,b} \left[\sum_i \max\{0, 1 - s_i(a^T v_i + b)\} + \lambda \|a\|^2 \right]$$

   (a) Write this problem as a quadratic program.

   (b) Derive the dual of the SVM problem.

3. **Adjustable optimization**

   In an uncertain optimization problem, there is some vector w that is only known initially to lie in the set $\mathcal{W} = \{w \mid Wx \leq 1\}$. The optimizer is allowed to select both of the following:

   - An initial decision x *before* w is realized.
   - A function $f : \mathcal{W} \to \mathbb{R}^n$ that produces an additional decision $z = f(w)$ *after* w is realized.

   In this problem assume that the function f is restricted to be *linear*, i.e. $f(w) = Mw$, and you are to choose the matrix M.

   Show that the following problem can be solved as a linear program:

   $$\begin{array}{ll} \underset{M,x}{\text{minimize}} & \max_{w \in \mathcal{W}} \left[c^\top (x + f(w)) + c_2^\top w \right] \\ \text{subject to} & A(x + f(w)) \leq b + Bw, \quad \forall w \in \mathcal{W} \end{array}$$

Algorithms

1. **Alternating Projection method for LPs**

   Consider a linear program in the standard form

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0, \end{array}$$

   with variable $x \in \mathbf{R}^n$, and where $A \in \mathbf{R}^{m \times n}$.

   (a) Find the dual of this problem. Use a vector ν as Langrange multipliers for the condition $Ax = b$, and $\lambda \geq 0$ as multipliers for the condition $x \geq 0$.

   (b) Write down the KKT conditions for the problem, and find a set of algebraic conditions in (x, ν, λ) whose solutions produce a primal-dual optimal solution using *only* linear equalities and inequalities. In particular, your conditions should *not* include any complementarity conditions.

   (c) Let $z = (x, \nu, \lambda)$ denote the collection of primal and dual decision variables. Express your optimality conditions as $z \in \mathcal{A} \cap \mathcal{C}$, where $\mathcal{A}$ is an affine set and $\mathcal{C}$ is a set of simple inequalities. Define appropriate F and g such that to write $\mathcal{A}$ as $\mathcal{A} = \{z \mid Fz = g\}$.

   (d) Explain how to compute the Euclidean projections onto $\mathcal{A}$ and $\mathcal{C}$. Your implementation should exploit *factorization caching* for the projection onto $\mathcal{A}$.

   (e) Implement an *alternating projection* method to solve the LP, i.e.

$$\begin{aligned} \tilde{z}^{k+1} &= \Pi_{\mathcal{A}}(z^k) \\ z^{k+1} &= \Pi_{\mathcal{C}}(\tilde{z}^{k+1}) \end{aligned}$$

   (f) Test your solver on a problem instance with $m = 100, n = 500$. Plot the residual $\|z^{k+1} - \tilde{z}^{k+1}\|_2$ over 1000 iterations. (This should converge to zero, although perhaps slowly.)

   Hint : Be careful to ensure that your randomly problems are actually feasible before solving them.

2. **Sparse signal recovery with Huber objective**

Suppose that you want to solve a *sparse signal recovery* problem as in the lectures, but with a Huber objective function ϕ as the penalty and a hard bound on the 1-norm of the weights x. In particular, you want to solve the problem

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \phi(Ax - y) \\ \text{subject to} & \|x\|_1 \leq N \end{array}$$

where ϕ is the Huber function

$$\phi(y) = \sum_i h(y_i), \qquad h(u) = \begin{cases} u^2 & |u| \leq 1 \\ (2|u| - 1) & |u| \geq 1 \end{cases}$$

Take N to be 10, and use the following Python code (or its equivalent in some other language of your choice) to generate A and y:

```
# Generate random signal
n = 1000
n_nnz = 10
hat_x = np.zeros(n)
nnz_idx = np.random.randint(0, n, n_nnz)
hat_x[nnz_idx] = np.random.randn(n_nnz)

# Samples
n_samples = 100
A = np.random.randn(n_samples, n)
y = A @ hat_x
```

Implement an algorithm to solve this problem using the projected gradient method.