EE127 Homework 6

Vighnesh Iyer

December 2, 2018

Exercise 1 (Safe feature elimination in sparse learning)

Consider the problem

$$p^* = \min_w \ f(X^T w) + \lambda \|w\|_1,$$

where $f : \mathbb{R}^m \to \mathbb{R}_p$ is convex, $X \in \mathbb{R}^{n,m}$ and $\lambda > 0$.
The above covers many sparse learning problems of interest, in which the function f is usually decomposable as a sum: $f(z) = \sum_{i=1}^m h(z_i)$ with h a convex univariate function, referred to as the loss function.
Note: In this problem, please do **not** assume that f is decomposable.
We denote by a_i^T the i-th row of X, $i = 1, \ldots, n$. These vectors correspond to the i-th variable in the learning problem. We seek a quick test to determine if we can safely remove variables in the learning problem, without affecting the optimal value. That is particularly helpful in the case of natural language processing studies where the number of features (i.e words) can scale up to a billion.
You may assume strong duality when needed.

1. Let us denote by f^* the so-called "conjugate" function of a function f, with values $f^*(v) = \max_z \ z^T v - f(z)$. Assume without proof that $(f^*)^* = f$. Show that a dual to the learning problem is

   $$d^* = \max_v \ -f^*(v) \ : \ \|Xv\|_\infty \leq \lambda.$$

   Hint: Rewrite the problem using the fact that $(f^*)^* = f$; then, express the problem as a maximization over one variable by adding constraints.

2. Now let v_0 be a dual feasible vector. Then we have $\|Xv_0\|_\infty \leq \lambda$. Let $\kappa := f^*(v_0)$. Show that, for a given $i \in \{1, \ldots, n\}$, the condition

   $$\lambda > \max_v \ |a_i^T v| \ : \ f^*(v) \leq \kappa$$

   allows to safely remove the i-th variable in the original problem. That is, we can reset w_i to zero in that problem, and get the same optimal value.

3. Show that the condition above can be expressed as

   $$\lambda > \lambda_i := \max \left(\min_{t \geq 0} \ \kappa t + t f(a_i/t), \min_{t \geq 0} \ \kappa t + t f(-a_i/t) \right).$$

   Explain why $\min_{t \geq 0} \ \kappa t + t f(a_i/t)$ and $\min_{t \geq 0} \ \kappa t + t f(-a_i/t)$ are convex problems.

   Note: This means the condition in part 2 can be checked very quickly, as it involves solving two one-dimensional convex problems. As you have shown in HW5, one-dimensional convex problems can be solved efficiently using bisection.

Solution:

1.

$$\begin{aligned}
f(v) = f^{**}(v) &= \max_z z^T v - f^*(z) \\
p^* &= \min_w \max_z z^T X^T w - f^*(z) + \lambda \|w\|_1 \\
&= \min_w \max_z z^T X^T w - f^*(z) + \max_{\|k\|_\infty \leq \lambda} k^T w \\
&= \min_w \max_{z, \|k\|_\infty \leq \lambda} -f^*(z) + z^T X^T w + k^T w \\
&= \min_w \max_{z, \|k\|_\infty \leq \lambda} -f^*(z) + (Xz + k)^T w \\
d^* &= \max_{z, \|k\|_\infty \leq \lambda} \min_w -f^*(z) + (Xz + k)^T w \\
&= \max_z -f^*(z) \quad : \quad \|Xz\|_\infty \leq \lambda
\end{aligned}$$

2. $-k = -f^*(v_0)$ is a lower bound on d^*. If a v is strictly feasible $\|Xv_0\|_\infty < \lambda$, then the condition $\lambda > |a_i^T v|$ holds. If v is maximized subject to $f^*(v) \leq k$ for the objective $\max_v |a_i^T v|$ and the objective is $< \lambda$, then it implies that a_i doesn't affect the optimal value of d^* since the constraint imposed by a_i isn't active at the optimum.

3.

$$\begin{aligned}
&\max_v |a_i^T v| \quad : \quad f^*(v) \leq k \\
&\max_v \min_t |a_i^T v| - (f^*(v) - k)t \\
&\max_v \min_t |a_i^T v| - (\max_z z^T v - f(z) - k)t \\
&\max_{z,v} \min_t |a_i^T v| - z^T v t + t f(z) + kt \\
&\max(\min_t t f(\frac{|a_i|}{t} + kt)) \\
&\max(\min_t t f(\frac{a_i}{t} + kt), \min_t t f(\frac{-a_i}{t} + kt))
\end{aligned}$$

The function kt is convex in t since it is linear, and the perspective function $tf(x/t)$ is convex, so the sum is also convex.

Exercise 2 (Retail pricing problem)

A retailer seeks to optimize the prices of n items, encoded in a vector $p \in \mathbb{R}^n_{++}$. For a given price vector $p \in \mathbb{R}^n$, the demand for the n items is modeled as an affine map. In economics, this linearity assumption is referred to as *elastic* demand. This demand is modeled by $d : \mathbb{R}^n \to \mathbb{R}^n$, with values $d_i(p) = d_i^0 - g_i(p_i - p_i^0)$ where p^0 is a reference price, d^0 is the corresponding demand, and $g \in \mathbb{R}^n_{++}$ is a vector of price "elasticities". We assume that the volume of sales always matches the demand. With this model, the total revenue is expressed as $r(p) = p^T d(p)$; we also define the *margin* (profit-to-revenue ratio) as the quantity $m(p) := \frac{(p-c)^T d(p)}{p^T d(p)}$ where $c \in \mathbb{R}^n_{++}$ is a vector containing the purchasing prices of the items. We seek to maximize the total revenue under several constraints:

- A lower bound on the margin, $m(p) \geq \beta$, where $\beta \in [0, 1]$ is given
- Inventory (storage) constraints on the demand, of the form $0 \leq d(p) \leq d^{\max}$, where $d^{\max} \in \mathbb{R}_{++}n$ is given
- Upper and lower bounds on p, of the form $p \in [p_l, p_u]$, with $0 \leq p_l \leq p_u$ given.

You may assume throughout the exercise that the problem is strictly feasible.

1. Express the problem as a convex problem, and label it with one acronym (LP, QP, QCQP, SOCP).

   Note: Choose the most constrained suitable problem formulation (e.g. if it can be expressed as LP, do not choose QCQP or SOCP). You do not need to prove that your formulation is the most general possible (e.g. you do not need to show formulating the problem as LP or QCQP is impossible).

2. Show that the problem can be equivalently written as the one-dimensional problem $\min_{\lambda \geq 0} D(\lambda)$ where D is a certain dual function which you will determine. Express $D(\lambda)$ as a convex problem over p.

3. Explain how to compute $D(\lambda)$ in $\mathcal{O}(n)$ time. What is the time complexity of computing $\lambda^* = \min_{\lambda \geq 0} D(\lambda)$? *Note:* If you choose to use bisection, please ignore logarithmic factors introduced (when computing time complexity) and disregard the problem of choosing an initial interval.

4. Detail how to recover an optimal primal point, once an optimal dual variable $\lambda^* = \min_{\lambda \geq 0} D(\lambda)$ is found. What is the time complexity of your process?

Solution:

1. The problem can be stated as:

$$\begin{aligned} \max_p r(p) &= \max_p \sum_{i=1}^n p_i(d_i^0 - g_i(p_i - p_i^0)) \\ p^* &= \max_p \sum_{i=1}^n -g_i p_i^2 + p_i(d_i^0 + g_i p_i^0) \\ -p^* &= \min_p \sum_{i=1}^n g_i p_i^2 - p_i(d_i^0 + g_i p_i^0) \end{aligned}$$

   The objective is quadratic and convex because $g_i > 0$.

   The constraints are:

   (a) $p_l \leq p_i \leq p_u$ for $i = 1, \ldots, n$, which is a linear box constraint

(b) $0 \leq d(p) \leq d^{max}$, which is an affine box constraint

(c) $m(p) \geq \beta \rightarrow r(p)(\beta - 1) + c^T d(p) \leq 0$, which is a convex quadratic constraint (since $r(p)$ is convex and $\beta - 1$ is negative)

Therefore, this is a QCQP.

2. We can dualize the quadratic constraint and obtain an optimization problem in the expected form:

$$\begin{aligned}
\mathcal{L}(p, \lambda) &= r(p) + \lambda(r(p)(\beta - 1) + c^T d(p)) \\
p^* &= \max_p \min_{\lambda \geq 0} r(p) + \lambda(r(p)(\beta - 1) + c^T d(p)) \quad : \quad p_l \leq p_i \leq p_u, 0 \leq d(p) \leq d^{max} \\
d^* &= \min_{\lambda \geq 0} \max_p r(p) + \lambda(r(p)(\beta - 1) + c^T d(p)) \quad : \quad p_l \leq p_i \leq p_u, 0 \leq d(p) \leq d^{max} \\
D(\lambda) &= \max_p r(p) + \lambda(r(p)(\beta - 1) + c^T d(p))
\end{aligned}$$

Because this problem is strictly feasible, and the original objective is concave over maximization, we can apply Slater's conditions, and find $d^* = p^*$. $D(\lambda)$ is convex over p since it is a maximization of a concave function.

3. $D(\lambda)$ can be computed by recognizing that each element of p, p_i can be maximized independently of any other element. So we can solve this set of 1-D maximization problems via bisection

$$\begin{aligned}
D(\lambda) &= \max_{p_i} \sum_{i=1}^{n} r(p_i) + \lambda(r(p_i)(\beta - 1) + c^T d_i(p_i)) \\
&= \sum_{i=1}^{n} \max_{p_i} p^T d_i(p_i) + \lambda(p^T d_i(p_i)(\beta - 1) + c^T d_i(p_i))
\end{aligned}$$

To do all these bisections to compute $D(\lambda)$ takes $\mathcal{O}(n)$ time. Solving for λ^* is just another bisection, and it will take $\mathcal{O}(log(\epsilon)) \rightarrow \mathcal{O}(1)$ time.

4. Using λ^*, plug it for λ in the d^* equation and find d^* already knowing p^*. Since Slater's condition holds, strong duality holds, and $p^* = d^*$. So the time complexity is $\mathcal{O}(1)$.

Exercise 3 (Optimal execution)

We are given a fixed number of shares $\bar{s}$ of a single asset, to be purchased over time intervals $t = 1, \ldots, T$. We denote by s_t the amount of shares to be purchased at time t, and refer to the vector $s = (s_1, \ldots, s_T) \in \mathbb{R}^T$ as our sequence of trades, so that $s^T \mathbf{1} = \bar{s}$, where $\mathbf{1} \in \mathbb{R}^T$ is a vector of ones. We treat s as a real vector (not an integer vector), and do not allow short selling, that is, we impose the constraint $s \geq 0$. We denote by p_t the price of the asset at time t, and refer to $p = (p_1, \ldots, p_T)$ as the price vector. The *execution cost* associated with a given sequence of trades $s \in \mathbb{R}^T_+$ is then $\sum_{t=1}^T p_t s_t$.
As we purchase s_t shares at each time t, $t = 1, \ldots, T$, the price p_t changes, not only due to (random) market dynamics, but also due to our purchases. A simple model for market impact dynamics is

$$p_t = p_{t-1} + \alpha s_t + r_t, \quad t = 0, \ldots, T, \tag{1}$$

where p_0, $\alpha > 0$ are model parameters, which we assume known. Here, the exogenous signal $r = (r_1, \ldots, r_T)$, which we also assume to be known for now, reflects the influence of the market as a whole on the price; for example, it may be derived from a simple (*e.g.*, auto-regressive) model for the SP 500 index. The market impact model above is simplistic as it does not guarantee positive prices, but we ignore that fact here.
Our goal is to find the best sequence of trades s so as to minimize the execution cost, subject to the constraints on s.

1. Show that we can write $p = A(\alpha s + r) + q$, where A a lower-triangular $T \times T$ matrix with 1's on the lower-triangular part, and $q \in \mathbb{R}^n$ is given.

2. Write the problem with decision variables s, p, and including the constraint (1). In that form, is the problem convex? Justify your answer carefully.

3. Write the problem as a QP in standard form. State precisely the variables and constraints. Make sure to check that the objective function is quadratic and convex in the variables of the problem.

   Hint: Show that $q(s) := s^T A s = s^T Q s$, with $Q := (1/2)(A + A^T)$, and that $2Q - I$ is PSD, with I the $T \times T$ identity matrix.

Solution:

1. A is a lower triangular matrix with 1s in the lower triangle and 0s in the upper triangle. q is a vector with all elements p_0.

$$A = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ 1 & 1 & 1 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$q = \begin{bmatrix} p_0 & p_0 & \ldots & p_0 \end{bmatrix}^T$$

2. The problem can be stated as:

$$\min_{s,p} \sum_{t=1}^T p_t s_t \quad : \quad s^T \mathbf{1} = \bar{s}, s \geq 0, p = A(\alpha s + r) + q$$

The first 2 constraints are a linear equality and a linear inequality. The final constraint can be expanded into multiple affine equalities:

$$\begin{aligned} p - \alpha As &= Ar + q \rightarrow \\ p_1 - s_1 &= r_1 + q_1 \\ p_2 - (s_1 + s_2) &= (r_1 + r_2) + q_2 \\ &\vdots \\ p_T - (s_1 + \cdots + s_T) &= r_T + q_T \end{aligned}$$

which are all convex equality constraints.

The objective isn't convex because the quadratic matrix has negative eigenvalues and thus isn't PSD. Here's how it looks (assuming $T = 2$):

$$\begin{aligned} x &= \begin{bmatrix} p_1 & \dots & p_T & s_1 & \dots & s_T \end{bmatrix}^T \\ p_1 s_1 + \cdots + p_T s_T &= x^T Q x \\ Q &= \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \end{bmatrix} \end{aligned}$$

Which has negative eigenvalues of -0.5, and isn't PSD. This structure generalizes for any T. This isn't a convex problem.

3. Write as a QP (by moving the equation for p into the objective)

$$\begin{aligned} &\min_{s,p} \sum_{t=1}^{T} p_t s_t \\ &= \min_{s,p} s^T p \\ &= \min_{s} s^T (A(\alpha s + r) + q) \\ &= \min_{s} \alpha s^T A s + s^T A r + s^T q \end{aligned}$$

The second and third terms are clearly linear in s and are convex. The first quadratic term can be analyzed further:

$$\begin{aligned} &s^T A s = s^T Q s \text{ for } Q = \frac{1}{2}(A + A^T) \\ &\text{because } s^T(\frac{1}{2}(A + A^T))s = \frac{1}{2} s^T A s + \frac{1}{2} s^T A^T s = s^T A s \\ &\text{then } 2Q - I \text{ is a matrix of all ones, which is trivially PSD} \end{aligned}$$

If $2Q - I$ is PSD then $2Q \geq I$ and Q is PSD; so the first term of the objective is convex quadratic. The same constraints $s^T \mathbf{1} = 1$ and $s \geq 0$ still apply which are linear, so this is a QP.

Exercise 4 (Hydro-electric power generation)

This exercise deals with the daily management of a set of hydro-electric plants in a valley. The goal is to balance the amount of water that is processed through different turbines that generate electricity, versus the amount that is conserved in reservoirs for future use. The planning takes place over a given time period, spanning typically a day, over which we plan to minimize cost.

Figure 1: A hydro-electric plant.

In this simplified version, we assume that the amount of water flowing into the reservoirs (due to rain or snow) is known in advance. The goal is to solve for the amounts processed through the turbines, to minimize a cost function. This function is assumed here to be fully known, although in practice it can depend on various factors, such as what happens with other electricity generation plants in the company's portfolio.
The optimization problem involves several time steps, denoted $k = 1, \ldots, K$; reservoirs, denoted $l = 1, \ldots, L$; hydro-electric plants, denoted $i = 1, \ldots, I$. Each plant may have several turbines, labelled with $j = 1, \ldots, J$, each with different rates of power generated versus volume of water processed.
We consider a specific problem with $K = 3$ time steps, with the topology given in Fig. 2.

Figure 2: A specific hydro-electric valley.

We use the following notation.

- $V_{l,0}$ is the initial (known) volume and $V_{l,K}$ is the final volume of reservoir l (expressed in $[m^3]$).
- The value of usage of the reservoir l is denoted as w_l (expressed in $[\$/m^3]$).
- λ_k represents the price for electric power paid at time k, expressed in [\$/MWh].
- ρ_j represents the efficiency of turbine j, expressed in $[MW/m^3]$.
- $T_{j,k}$ is the volume of water going through turbine j at time k (in $[m^3]$).

Let V be an $L \times K$ matrix and T a $J \times K$ matrix that represent respectively the volume of reservoirs and the volume of water going through the turbines at different timesteps.
We seek to minimize the overall cost of production. The objective function F reflecting the arbitration between immediate use of water through a turbine, or water conservation for later use, is:

$$\begin{aligned} F(V,T) = & \sum_{l=1}^{L} w_l(V_{l,0} - V_{l,K}) - \sum_{k=1}^{K}\sum_{j=1}^{J} \lambda_k \rho_j T_{j,k} \\ = & \sum_{l=1}^{L} w_l V_{l,0} - \left(\sum_{l=1}^{L} w_l V_{l,K} + \sum_{k=1}^{K}\sum_{j=1}^{J} \lambda_k \rho_j T_{j,k} \right) \end{aligned}$$

The constraints are as follows.

- The evolution of the volumes of the reservoirs must meet a flow equation:

$$V_{l,k} = V_{l,k-1} + A_{l,k} + \sum_{j \in U_l} T_{j,k-d_j^l} - \sum_{j \in D_l} T_{j,k+d_j^l}$$

  where

- $j \in U_l$: turbine j upstream of the reservoir l;
- $j \in D_l$: turbine j downstream of the reservoir l;
- d_j^l: the delay of water flow inside the plant, from upstream turbine if $j \in U$ and to downstream turbine if $j \in D$. For this exercise, we ignore the delay, i.e. $d_j^l = 0$;
- $A_{l,k}$: water inflow (e.g. rain, melt snow) to reservoir l at time k.

- We have bounds on the volumes of water going through turbines:

$$T_{j,k}^{\min} \leq T_{j,k} \leq T_{j,k}^{\max}, \ \ \forall\, k, j.$$

  For simplicity, we will set $T_{j,k}^{\min} = 0$.

- We have bounds on the volumes of reservoirs:

$$V_{l,k}^{\min} \leq V_{l,k} \leq V_{l,k}^{\max}, \ \ \forall\, k, l.$$

1. Consider the valley topology in Fig. 2 (notations are different from the one in the text). In the figure, let reservoir 1 and 2 refer to the upstream and downstream reservoirs, respectively. Furthermore, let turbines 1 and 2 refer to the groups G_1, G_2 in plant 1, and turbine 2 refer to the group G_3 in plant 2.

   Show that the flow equations are:

   - Time $k = 1$:
     Reservoir $l = 1$: $V_{11} = V_{10} + A_{11} - (T_{11} + T_{21})$,
     Reservoir $l = 2$: $V_{21} = V_{20} + A_{21} + (T_{11} + T_{21}) - T_{31}$.
   - Time $k = 2$:
     Reservoir $l = 1$: $V_{12} = V_{11} + A_{12} - (T_{12} + T_{22})$,
     Reservoir $l = 2$: $V_{22} = V_{21} + A_{22} + (T_{12} + T_{22}) - T_{32}$.
   - Time $k = 3$:
     Reservoir $l = 1$: $V_{13} = V_{12} + A_{13} - (T_{13} + T_{23})$,
     Reservoir $l = 2$: $V_{23} = V_{22} + A_{23} + (T_{13} + T_{23}) - T_{33}$.

2. Solve for the optimal $\hat{T}$ when $\lambda = (10, 30, 8)^T$. Use the parameters loaded for you in the notebook.

3. We are now due to solve a new problem with slightly different input parameters. Due to contractual constraints, the new solution T should not differ from the previous configuration $\hat{T}$. Precisely, we would like to limit the *number* of turbines that would be affected by a change by restricting the number of rows in the matrix T that differ from the corresponding rows in $\hat{T}$. Formally, construct a convex proxy $R(T - T')$, that penalizes the sum of the absolute values of maximum change in turbine flow between the rows of T and T'. Use a regularization parameter of $\gamma = 10^{10}$ in your implementation—so the entire term added to the objective is $\gamma R(T - T')$.

   Hint: Define a convex proxy function for the number of changes, with a regularization parameter that you can take to be $\gamma = 10^{10}$.

4. Test your heuristic with the data loaded for you in part 2 of the notebook:

Solution:

1. In this case we have 2 reservoirs (l_1, l_2), 3 turbines (j_1, j_2, j_3) of which j_1, j_2 and j_3 are downstream from reservoir l_1 and l_2 respectively. On generic timestep k:

$$V_{1,k} = V_{1,k-1} + A_{1,k} - (T_{1,k} + T_{2,k})$$
$$V_{2,k} = V_{2,k-1} + A_{2,k} + (T_{1,k} + T_{2,k}) - T_{3,k}$$

   And expanding these out gives us the flow equations in the problem statement.

2.
```
rho_tiled = np.reshape(np.repeat(rho, K), (J, K))
lam_tiled = np.tile(lam, (J, 1))
rho_lam = np.multiply(rho_tiled, lam_tiled)
V = cvx.Variable((L,K))
T = cvx.Variable((J,K))
obj = cvx.Minimize(V[:,0] * w_l - V[:,K-1] * w_l - cvx.sum(cvx.multiply(rho_lam, T)))
constraints = [T >= np.zeros((J,K)), T <= T_max, V >= V_min, V <= V_max,
               V[0, 0] == V_0[0] + A[0, 0] - (T[0, 0] + T[1, 0]),
               V[1, 0] == V_0[1] + A[1, 0] + (T[0, 0] + T[1, 0]) - T[2, 0],
               V[0, 1] == V[0, 0] + A[0, 1] - (T[0, 1] + T[1, 1]),
               V[1, 1] == V[1, 0] + A[1, 1] + (T[0, 1] + T[1, 1]) - T[2, 1],
               V[0, 2] == V[0, 1] + A[0, 2] - (T[0, 2] + T[1, 2]),
               V[1, 2] == V[1, 1] + A[1, 2] + (T[0, 2] + T[1, 2]) - T[2, 2]
              ]
prob = cvx.Problem(obj, constraints)
prob.solve()
obj_val = prob.value
T_opt = np.array(T.value)

# Round values for readability (optional)
obj_val = np.around(obj_val, decimals=7)
T_opt = np.around(T_opt, decimals=0)

Optimal value:
-4117.4826773

Optimal water volumes, T_opt:
[[ 0. 55.  0.]
 [30. 65. 15.]
 [80. 80. -0.]]
```

3. The goal is to encourage sparsity in the vector $\|T_i - \hat{T}_i\|_2$, which is evaluated on rows $i = 1, \ldots, J$. So put the L1 norm of that vector as a penalty term in the objective.

$$R(X) = \|\|X_i - \hat{X}_i\|_2 \quad \forall i = 1, \ldots, J\|_1$$

4. The new objective became:

```
obj = cvx.Minimize(V[:,0] * w_l - V[:,K-1] * w_l - cvx.sum(cvx.multiply(rho_lam, T)) +
                   gamma * cvx.sum(
                       [cvx.norm(T[j] - T_hat[j], 2) for j in range(J)]
                   ))
```

With $\gamma = 10^{10}$ I didn't see any change in $T_{opt,reg}$. I had to lower to $\gamma = 1$ see the function minimized further and a different result, with a changed second row.

```
Optimal value:
-4244.1110298

Optimal water volumes, T_opt_reg:
[[ 0. 55.  0.]
```

```
[30. 65.  0.]
[80. 80. -0.]]
```

Exercise 5 (Robust Machine Learning)

We consider a binary classification problem, where the prediction label associated with a test point $x \in \mathbb{R}^n$, is the form $\hat{y}(x) = \mathbf{sign}(w^T x + v)$, with $(w, v) \in \mathbb{R}^m \times \mathbb{R}$ the classifier weights. Given a training set X, y, with $X = [x_1, \ldots, x_m] \in \mathbb{R}^{n \times m}$ the data matrix, with data points $x_i \in \mathbb{R}^n$, $i = 1, \ldots, m$, and $y \in \{-1, 1\}^m$ the vector of corresponding labels, the training problem is to minimize the so-called hinge loss function:

$$\min_{w,v} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w + v)). \tag{2}$$

We seek to find a classifier (w, v) that can be implemented with low precision (say, as an integer vector). To this end, we modify the training problem so that it accounts for the implementation error, when approximating the original optimal (full precision) weight vector w_* with a low-precision one, $\tilde{w}$. We bound the corresponding error as $\|\tilde{w} - w_*\|_\infty \leq \epsilon$ for some given absolute error bound $\epsilon > 0$; for example, if $\tilde{w}$ is the nearest integer vector, the error is bounded by $\epsilon = 0.5$. Then, we seek to solve the *robust counterpart* to (2):

$$\min_{w,v} \max_{\substack{\tilde{w} \,:\, \|\tilde{w} - w\|_\infty \leq \epsilon \\ \tilde{v} \,:\, |\tilde{v} - v| \leq \epsilon}} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T \tilde{w} + \tilde{v})). \tag{3}$$

1. Justify the use of the hinge loss function in problem (2); in particular, explain geometrically what it means to have a zero loss.

2. Show that without loss of generality, we can reformulate this problem as a robust optimization over the variable w only, which we will do henceforth.

   Hint: Think about modifying the data vectors x_i.

3. Explain how to obtain a low-precision classifier once problem (3) is solved. What guarantees do we have on the training error?

4. Show that the optimal value of problem (3) is bounded above by

$$\min_{w} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w) + \epsilon \|x_i\|_1). \tag{4}$$

5. Assume that the data set is normalized, in the sense that $\|x_i\|_1 = 1$, $i = 1, \ldots, m$. How would you solve problem (4) if you had code to solve (2) only?

Solution:

1. The lower bound of the hinge-loss function is 0, which is attained when for every data point $y_i(x_i^T \omega + v) \geq 1$, which begins to hold when the classifier is perfect and the weight and bias has been adjusted to make the product ≥ 1. Geometrically this means that the half spaces $\omega^T x + v \leq -1$ and $\omega^T x + v \geq 1$ contain all the points in each of their respective categories.

2. This can be done by adding a new variable to each x_i datapoint with value 1 and adding another

variable to ω. Formally:

$$
\begin{aligned}
&x_i' = [x_{i,0}, \ldots, x_{i,n}, 1] \quad \forall i = 1, \ldots, m \\
&\omega' = [\omega_0, \ldots, \omega_n, v] \\
&X' \in \mathbb{R}^{n+1 \times m} \\
&\omega' \in \mathbb{R}^{n+1} \\
&\min_{\omega'} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i'^T \omega'))
\end{aligned}
$$

3. After finding w and from the robust optimization problem, round up each element of w to the nearest multiple of $2 * \epsilon$. For example, if $\epsilon = 0.5$, 0.1 would be rounded to 0, and 0.6 would be rounded to 1.0. The training error of the low-precision w vector will be no worse than the solution to the robust optimization problem since it has considered the worst case deviation values for $\tilde{w}$.

4. Expand the inner maximization constraint:

$$
\begin{aligned}
&\max_{\tilde{w}: \|\tilde{w} - w\|_\infty \leq \epsilon} \rightarrow \\
&\max(|\tilde{w}_i - w_i|) \leq \epsilon \quad \forall i = 1, \ldots, n+1 \\
&|\tilde{w}_1 - w_1| \leq \epsilon \\
&\ldots \\
&|\tilde{w}_{n+1} - w_{n+1}| \leq \epsilon
\end{aligned}
$$

Take the worst case where every $|\tilde{w}_i - w_i| = \epsilon$, then write the objective's upper bound:

$$
\begin{aligned}
&\leq \min_w \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T(w \pm \epsilon)) \\
&\min_w \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w \pm x_i \epsilon)) \\
&\min_w \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w) \pm y_i x_i \epsilon)) \\
&\min_w \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w) + \max_{y_i: \|y_i\|_\infty \leq 1} y_i x_i \epsilon)) \\
&\min_w \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w) + \|x_i\|_1 \epsilon))
\end{aligned}
$$

5. Augument each data point x_i with another variable with fixed value $-\frac{\epsilon}{y_i}$ for the y_i associated with that x_i. Add another entry to w for the new variable in each data point x_i.

$$
\begin{aligned}
&x_i' = [x_1, \ldots, x_n, -\frac{\epsilon}{y_i}] \\
&w' = [w_1, \ldots, w_n, w_{n+1}]
\end{aligned}
$$

Solve the problem (2) using x_i', w'. To recover the solution to (4) normalize the solution w_{opt} by dividing it by $|w_{opt,n+1}|$ and removing the last element.

$$w_{opt,problem4} = \frac{w_{opt,problem2}}{w_{opt,problem2,n+1}}$$