EE 127 / EE 227AT 11/13/18
L. El Ghaoui
A. Bayen

# HW #6 Solutions

**Exercise 1 (Safe feature elimination in sparse learning)**

Consider the problem

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

where $f : \mathbb{R}^m \to \mathbb{R}_+$ 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 $a_i^\top$ 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 $f^*(v) = \max_z \ z^\top v - f(z)$. 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

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

   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

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

Explain why $\min_{t\geq 0}\ \kappa t + tf(a_i/t)$ and $\min_{t\geq 0}\ \kappa t + tf(-a_i/t)$ 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**

1. Replacing $f = (f^*)^*$ and $\lambda\|w\|_1 = \lambda\| - w\|_1 = \max_{u:\|u\|_\infty\leq\lambda} -u^T w$, we have:

$$\begin{aligned} p^* &= \min_w \max_{u,v}\ v^\top X^\top w - f^*(v) - u^\top w\ :\ \|u\|_\infty \leq \lambda \\ d^* &= \max_{u,v} \min_w\ v^\top X^\top w - f^*(v) - u^\top w\ :\ \|u\|_\infty \leq \lambda \\ &= \max_{u,v} \min_w\ (Xv-u)^\top w - f^*(v)\ :\ \|u\|_\infty \leq \lambda. \end{aligned}$$

It follows that:

$$\min_w\ (Xv-u)^\top w - f^*(v) = \begin{cases} -\infty & \text{if } Xv \neq u \\ -f^*(v) & \text{otherwise.} \end{cases}$$

This gives us:

$$\begin{aligned} d^* &= \max_{u,v} -f^*(v)\ :\ Xv = u, \|u\|_\infty \leq \lambda \\ &= \max_v -f^*(v)\ :\ \|Xv\|_\infty \leq \lambda \end{aligned}$$

as desired.

2. First, note that strong duality holds, using Sion's theorem and the convexity of $f^*$. Therefore, $p^* = d^*$. The constraints of the dual problem write

$$|a_i^\top v| \leq \lambda,\ \ i = 1, \ldots, n.$$

Since $v_0$ is dual feasible, we have $d^* \geq -f^*(v_0) = -\kappa$, which implies that $f^*(v) \leq f^*(v_0) = \kappa$ for any optimal $v$. If, for a given $i \in \{1, \ldots, n\}$, we have

$$|a_i^\top v| < \lambda \text{ whenever } f^*(v) \leq \kappa,$$

then the $i$-th constraint is not active in the dual at optimum, and can be safely removed. Removing the $i$-th constraint is equivalent to setting the $i$-th row of $X$ to zero; the optimal value and the optimal $\nu^*$ will not be affected. Retracing the steps in part 1 to convert the modified dual problem back into primal form, we see that this is equivalent to to the primal problem with the $i$-th row of $X$ set to zero. Therefore, we can safely set $w_i$ to zero without affecting the optimal value.

3. The condition reads $\lambda > \max(\phi(a_i), \phi(-a_i))$, where for a given $a \in \mathbb{R}^n$:

$$\begin{aligned}
\phi(a) &:= \max_v \; a^\top v \;:\; f^*(v) \leq \kappa \\
&= \max_v \min_{t \geq 0} \; a^\top v + \kappa t - t f^*(v) \\
&= \min_{t \geq 0} \max_v \; a^\top v + \kappa t - t f^*(v) \\
&= \min_{t \geq 0} \; \kappa t + t \max_v \; \big( (a/t)^\top v - f^*(v) \big) \\
&= \min_{t \geq 0} \; \kappa t + t f(a/t).
\end{aligned}$$

We further have that the function $t > 0 \to t f(a/t)$ is a convex function, since it can be expressed as a point-wise maximum of functions linear in $t$. Using $(f^*)^* = f$,

$$\begin{aligned}
t f(a/t) &= t \cdot \max_z \; z^\top (a/t) - f^*(z) \\
&= \max_z \; z^\top a - t f^*(z),
\end{aligned}$$

where we see that for every $z$, $(a, t) \to z^\top a - t f^(z)$ is linear, hence convex.

The minimization problems involved are thus 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 2**

1. Define

$$R(p) := p^T d(p)$$
$$M(p) := ((1-\beta)p - c)^T d(p).$$

Note that $R, M$ are concave in $p$. To see this, let $G = \operatorname{diag}(g)$ and expand $d(p)$ in $R(p), M(p)$ to get

$$R(p) = -p^T G p + (d^0 - p^0)^T p$$
$$M(p) = -(1-\beta)p^T G p + \left[(1-\beta)(d^0 - p^0) + G^T c\right]^T p + (c^T G p^0 - c^T d)$$

We can formulate the optimization problem as

$$\max_p \ R(p) \ : M(p) \geq 0, \ \ 0 \leq d(p) \leq d^{\max}, \ \ p_l \leq p \leq p_u.$$

Since both $R, M$ are concave functions of $p$, the problem is convex. Substituting $R(p), M(p)$ from above, we see the problem is a QCQP.

2. The only "coupling" constraint in the problem is the margin constraint, which we dualize. For $\lambda \geq 0$ we define the corresponding dual function

$$D(\lambda) := \max_{p \,:\, p_l \leq p \leq p_u} R(p) + \lambda M(p) \ : \ 0 \leq d(p) \leq d^{\max}.$$

$R, M$ are concave in $p$, and so $D$ can be expressed as a convex optimization problem in $p$. Assuming the original problem is strictly feasible, by strong duality, we have $p^* = \min_{\lambda \geq 0} \ D(\lambda)$.

3. We can express $D(\lambda)$ as the sum of $n$ one-dimensional optimization problems:

$$\begin{aligned} D(\lambda) &= \max_{p \,:\, p_l \leq p \leq p_u} R(p) + \lambda M(p) \ : \ 0 \leq d(p) \leq d^{\max} \\ &= \max_{p \,:\, p_l \leq p \leq p_u} (1 + \lambda(1-\beta))p^T d(p) - \lambda c^T d(p) \ : \ 0 \leq d(p) \leq d^{\max} \\ &= \sum_{i=1}^{n} \max_{p_i \,:\, p_{l,i} \leq p_i \leq p_{u,i}} (1 + \lambda(1-\beta))p_i d_i(p) - \lambda c_i d_i(p) \ : \ 0 \leq d_i(p) \leq d_i^{\max}. \end{aligned}$$

Each of these one-dimensional optimization problems can be solved efficiently using bisection. Hence, $D(\lambda)$ can be computed in $\mathcal{O}(n)$ time (ignoring logarithmic factors). We can now solve for the one-dimensional dual problem $\lambda^* = \min_{\lambda \geq 0} D(\lambda)$, again using bisection, in $\mathcal{O}(n)$ time (ignoring logarithmic factors).

4. Once an optimal point $\lambda^*$ is obtained, we can recover an optimal point as

$$p^* = \arg\max_p \ R(p) + \lambda^* M(p).$$

The objective is quadratic and the problem is unconstrained, so we can solve for a closed form solution for $p^*$. The time complexity of this process is $\mathcal{O}(n)$ since $G$ is diagonal.

**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 $s^\top \mathbf{1} = \bar{s}$, 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 $q(s) := s^\top A s = s^\top Q s$, with $Q := (1/2)(A + A^\top)$, and that $2Q - I$ is PSD, with $I$ the $T \times T$ identity matrix.

**Solution 3**

1. We have

$$p_t = p_0 + \sum_{i=1}^{t} (\alpha s_i + r_i),$$

   and so

$$p = A(\alpha s + r) + q,$$

where

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}, \quad q = \begin{pmatrix} p_0 \\ p_0 \\ \vdots \\ p_0 \end{pmatrix}.$$

2. The problem writes

$$\min_{s,p} \; p^\top s \;:\; s \geq 0, \;\; s^\top \mathbf{1} = \bar{s}, \;\; p = A(\alpha s + r) + q. \tag{2}$$

As such it is not a QP. To check this, we observe that the objective function, $f : (p, s) \to p^\top s$, is quadratic, and contains no linear of constant terms (it is a "quadratic form"). Quadratic forms are convex if and only if they are non-negative everywhere. This is not true, as the special case $p = -s$ reveals.

3. Problem (2) is not convex. However, after we eliminate $p$, the problem becomes convex.

Indeed, the execution costs are

$$s^\top p = s^\top (A(\alpha s + r) + q) = \alpha s^\top A s + s^\top (Ar + q).$$

Since $\alpha > 0$ it suffices to show that the quadratic function $q : \mathbb{R}^m \to \mathbb{R}$ with values $q(s) := s^\top A s$ is convex. We use the hint:

$$q(s) = s^\top A s = \frac{1}{2}(s^\top A s + s^\top A^\top s) = s^\top Q s,$$

With $Q := (1/2)(A + A^\top)$. It is readily verified that the diagonal elements of $2Q$ are all 2's and off-diagonal elements are all 1's:

$$2Q = A + A^\top = \begin{pmatrix} 2 & 1 & \dots & 1 \\ 1 & 2 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 2 \end{pmatrix} = I + \mathbf{1}\mathbf{1}^\top.$$

Hence the form $q$ is positive semi-definite:

$$\text{For every } s\text{: } q(s) = s^\top Q s = \frac{1}{2}(s^\top s + (\mathbf{1}^\top s)^2) \geq 0.$$

Hence the matrix $Q$ is positive semi-definite (PSD), and the associated quadratic function $q$ is convex.

To summarize, our problem writes

$$\min_{s} \; \alpha s^\top Q s + s^\top (Ar + q) \;:\; s \geq 0, \;\; s^\top \mathbf{1} = \bar{s}.$$

This is a QP (since $\alpha > 0$, $Q$ PSD).

**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.

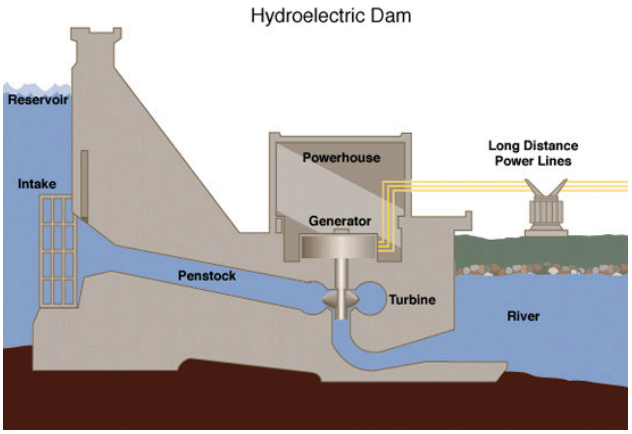


Figure 1: A hydro-electric plant.

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.

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 $[\$/m^3]$).

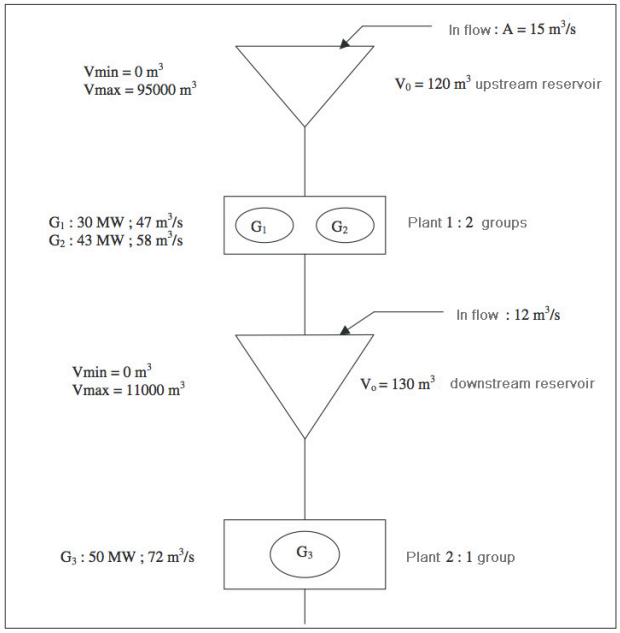


Figure 2: A specific hydro-electric valley.

- $\lambda_k$ represents the price for electric power paid at time $k$, expressed in [\$/MWh].
- $\rho_j$ 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 $\lambda = (10, 30, 8)^\top$. 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. This means that the number of rows in matrix $T$ that differ

from the corresponding one in $\hat{T}$ should not be big. Suggest an efficient heuristic for this using several options of norms like $\|\cdot\|_\infty, \|\cdot\|_1, \|\cdot\|_2$ applied on the matrix $T - \hat{T}$ either as a whole or in combinations on different axis. *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 4**

1. The optimization problem is of the form:

$$\begin{aligned} \text{minimize} \quad & \sum_{l=1}^{L} w_l V_{l,0} - \left( \sum_{l=1}^{L} w_l V_{l,K} + \sum_{k=1}^{K} \sum_{j=1}^{J} \lambda_k \rho_j T_{j,k} \right) \\ \text{subject to} \quad & V_{l,k} = V_{l,k-1} + A_{l,k} + \sum_{j \in U_l} T_{j,k} - \sum_{j \in D_l} T_{j,k}, \forall j, k, \\ & T_{j,k}^{\min} \leq T_{j,k} \leq T_{j,k}^{\max}, \forall j, k, \\ & V_{l,k}^{\min} \leq V_{l,k} \leq V_{l,k}^{\max}, \forall l, k.t \end{aligned}$$

As an example, at the time $k = 1$ we obtain the following equations:
Reservoir $l = 1$: $V_{11} = V_{10} + A_{11} - (T_{11} + T_{21})$
This is because the volume in reservoir $l = 1$ will be the sum of the volume already existing from the previous timestep $k = 0$ and the inflow to reservoir 1, and we subtract the volume passing through the 2 turbines in the reservoir.

Reservoir $l = 2$: $V_{21} = V_{20} + A_{21} + (T_{11} + T_{21}) - T_{31}$
In reservoir 2, there is only 1 turbine. Again, we add the existing amount, and the inflow at time 1, which includes the incoming flow from the 2 turbines upstream. Then, we subtract the outflow from the 3$^{\text{rd}}$ turbine.
We can repeat this for other time steps to get the other flow equations

2. The optimal $\hat{T}$ we get is

$$\hat{T} = \begin{bmatrix} 0 & 55 & 0 \\ 30 & 65 & 15 \\ 0 & 80 & 0 \end{bmatrix}$$

See the solution notebook for the code.

3. Define the following convex proxy for the number of changes:

$$R(T) \doteq \sum_{j=1}^{J} \max_{1 \leq k \leq K} |T_{j,k} - \hat{T}_{j,k}|.$$

We add this function as regularization term to the objective function, as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{l=1}^{L} w_l V_{l,0} - \left( \sum_{l=1}^{L} w_l V_{l,K} + \sum_{k=1}^{K} \sum_{j=1}^{J} \lambda_k \rho_j T_{j,k} \right) + \gamma R(T) \\
\text{subject to} \quad & V_{l,k} = V_{l,k-1} + A_{l,k} + \sum_{j \in U_l} T_{j,k} - \sum_{j \in D_l} T_{j,k}, \forall l, k, \\
& T_{j,k}^{\min} \leq T_{j,k} \leq T_{j,k}^{\max}, \forall j, k, \\
& V_{l,k}^{\min} \leq V_{l,k} \leq V_{l,k}^{\max}, \forall l, k,
\end{aligned}$$

where $\gamma > 0$ should be chosen so that the desired number of changes is achieved.

4. The optimal $\hat{T}$ we get after regularization using $10^{10}$ is

$$\hat{T} = \begin{bmatrix} 0 & 12 & 0 \\ 5 & 10 & 0 \\ 0 & 56 & 0 \end{bmatrix}$$

**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:

$$\min_{w,v} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w + v)). \tag{3}$$

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 (3):

$$\min_{w,v} \max_{\substack{\tilde{w} \,:\, \|\tilde{w} - w\|_\infty \leq \epsilon \\ \tilde{v} \,:\, |\tilde{v} - v| \leq \epsilon}} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T \tilde{w} + \tilde{v})). \tag{4}$$

1. Justify the use of the hinge loss function in problem (3); 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 (4) is solved. What guarantees do we have on the training error?

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

$$\min_{w} \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y_i(x_i^T w) + \epsilon \|x_i\|_1). \tag{5}$$

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 (5) if you had code to solve (3) only?

**Solution 5**

1. The hinge loss function is an upper bound on the so-called $0-1$ error loss,

$$\frac{1}{m}\sum_{i=1}^{m} E(y_i(x_i^T w + v)),$$

where $E$ is the function with values $E(\xi) = 1$ if $\xi < 0$, 0 otherwise. An alternate justification is that the function minimizes the distance of wrongly classified points to the decision boundary, which is the hyperplane $\mathcal{H}$ described by the equation $w^T x + v = 0$.

If the loss is zero, we have

$$\forall\, i = 1, \ldots, m \;:\; y_i(x_i^T w + v) \leq 0,$$

which means geometrically that the hyperplane $\mathcal{H}$ separates the positive and negative classes.

2. We can simply append a 1 at the end of each data point $x_i$, $i = 1, \ldots, m$.

3. Once $w^*$ is found, we simply replace it by its closest low-precision approximation $\tilde{w}$. Since that approximation satisfies the bound $\|\tilde{w} - w_*\|_\infty \leq \epsilon$, we know that the training error corresponding to the low-precision classifier:

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

is no worse than the worst-case training error, that is, the optimal value of the robust problem (4).

4. We have

$$\max_{\delta\,:\,\|\delta\|_\infty \leq \epsilon} \delta^T z = \epsilon \cdot \sum_{j=1}^{n} \max_{\delta_j\,:\,|\delta_j| \leq 1} \delta_j z_j = \epsilon \|z\|_1.$$

Therefore

$$\max_{\|\delta\|_\infty \leq \epsilon} \max(0, 1 - y_i(x_i^T(w + \delta)) = \max(0, 1 - y_i x_i^T w + \epsilon \|x_i\|_1). \tag{6}$$

We now turn to the full loss function. Since the max function is convex, the maximum of a sum is less than the sum of maxima, resulting in

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

which, in view of (6), leads to the desired result.

5. When the data is normalized, problem (5) reads

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

We can divide each term by $1+\epsilon$, and set $\bar{w} = \frac{1}{1+\epsilon} w$; the new problem reads just as (3).