CS 446 / ECE 449 — Homework 3

yiminc2

Instructions.

- Homework is due **Friday, October 17**, at 11:59 **PM** CST; you have **3** late days in total for **all Homeworks**.
- Everyone must submit individually at gradescope under `Homework 3` and `Homework 3 Code`.
- The "written" submission at `Homework 3` **must be typed**, and submitted in any format gradescope accepts (to be safe, submit a PDF). You may use LaTeX, markdown, google docs, MS word, whatever you like; but it must be typed!
- When submitting at `Homework 3`, gradescope will ask you to **mark out boxes around each of your answers**; please do this precisely!
- Please make sure your NetID is clear and large on the first page of the homework.
- Your solution **must** be written in your own words. Please see the course webpage for full **academic integrity** information. You should cite any external reference you use.
- We reserve the right to reduce the auto-graded score for `Homework 3 Code` if we detect funny business (e.g., your solution lacks any algorithm and hard-codes answers you obtained from someone else, or simply via trial-and-error with the autograder).
- When submitting to `Homework 3 Code`, upload `hw3_q2.py`, `hw3_q4.py`, and `hw3_utils.py`. Additional files will be ignored.

1. Support Vector Machines. (25 pt)

Recall that, for a soft-margin SVM, we assume the optimization objective is

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \quad s.t.\ y^{(i)}(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i \in \{1, 2, ..., N\}$$

(a) **Soft margin with hinge loss. (7 pt)**
Use the dataset in $\mathbb{R}^2$:

$$x^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ y^{(1)} = +1; \quad x^{(2)} = \begin{bmatrix} 2 \\ 0 \end{bmatrix},\ y^{(2)} = +1; \quad x^{(3)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ y^{(3)} = -1.$$

For any $(\boldsymbol{w}, b)$ define the functional margin $\gamma_i := y^{(i)}(\boldsymbol{w}^\top x^{(i)} + b)$, the hinge loss / slack $\xi_i := \max\{0,\ 1 - \gamma_i\}$.

(i) With $\boldsymbol{w} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $b = 0$, $C = 1$, compute: (2 pt)

- $f(x^{(i)}) := \boldsymbol{w}^\top x^{(i)} + b$
- γ_i for each $x^{(i)}$,
- ξ_i for each $x^{(i)}$
- and the objective value.

(ii) With $\boldsymbol{w} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $b = 0$, $C = 1$, repeat the computations in (i). Which parameter choice has the smaller objective value? (2 pt)

(iii) Re-evaluate the objective value for both (i) and (ii) with $C = 0.5$ and with $C = 2$. Briefly describe how increasing C and decreasing C change the trade-off between margin size and training violations. (3 pt)

(b) **Importance weighted soft margin SVMs. (18 pt)**
You are given a training dataset $\{(x^{(i)}, y^{(i)}, p^{(i)})\}_{i=1}^{N}$ where $y^{(i)} \in \{-1, +1\}$ and $0 \leq p^{(i)} \leq 1$ is the importance weight of i-th point.

i. Write the **primal** optimization in which each example's slack penalty is scaled by $p^{(i)}$. (3 pt)

ii. Derive the **dual** problem. Show how the weight $p^{(i)}$ changes the feasible set for the dual variables α_i. (6 pt)

iii. Suppose we have three training samples with $p^{(1)} = 1$, $p^{(2)} = \frac{1}{2}$, $p^{(3)} = 0$. Suppose $C = 2$, what are the feasible sets for $\alpha_1, \alpha_2, \alpha_3$? (3 pt)

iv. We now assume that the optimization objective for this L_2 soft-margin SVM is

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \quad s.t.\ y^{(i)}(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i \in \{1, 2, ..., N\}$$

Derive the dual problem. (6 pt)

(a) i. With $w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $b = 0, C = 1$ compute:

- $f(x^{(i)}) = w^\intercal x^{(i)} + b$
  - A. $f(x^{(1)}) = 0$
  - B. $f(x^{(2)}) = 2$
  - C. $f(x^{(3)}) = 1$
- γ_i for each $x^{(i)}$

A. $\gamma_1 = 0$
B. $\gamma_2 = 2$
C. $\gamma_3 = -1$

- ξ_i for each $x^{(i)}$
  A. $\xi_1 = 1$
  B. $\xi_2 = 0$
  C. $\xi_3 = 2$
- Objective Value
  Ans: 3.5
  $\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} \xi_i$
  $= \frac{1}{2} + 3$
  $= 3.5$

ii. With $w = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad b = 0, \quad C = 1$

Ans: $w = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad b = 0, \quad C = 1$

- $f(x^{(i)}) = w^\intercal x^{(i)} + b$
  A. $f(x^{(1)}) = 0$
  B. $f(x^{(2)}) = 2$
  C. $f(x^{(3)}) = 0$
- γ_i for each $x^{(i)}$
  A. $\gamma_1 = 0$
  B. $\gamma_2 = 2$
  C. $\gamma_3 = 0$
- ξ_i for each $x^{(i)}$
  A. $\xi_1 = 1$
  B. $\xi_2 = 0$
  C. $\xi_3 = 1$
- Objective Value
  $\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} \xi_i$
  $= \frac{1}{2} + 2$
  $= 2.5$

$\because 2.5 < 3.5 \therefore w = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad b = 0, \quad C = 1$

iii. **Ans: Increasing C leads to $\xi_i = 0$ and induces smaller margin, whereas decreasing C tolerates larger ξ_i for misclassifications, giving rise to larger margin. When $C = \infty$, the SVM becomes hard-margin.**

- $w = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad b = 0$
  - $C = 0.5$
    $\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} \xi_i$
    $= \frac{1}{2} + 0.5 \times 3$
    $= 2$
  - $C = 2$
    $\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} \xi_i$
    $= \frac{1}{2} + 2 \times 3$
    $= 6.5$

- $w = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad b = 0, \quad C = 0.5$
  - $C = 0.5$
    $\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} \xi_i$
    $= \frac{1}{2} + 0.5 \times 2$
    $= 1.5$
  - $C = 2$
    $\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} \xi_i$
    $= \frac{1}{2} + 2 \times 2$
    $= 4.5$

(b) i. Write the primal optimization in which each example's slack penalty is scaled by $p^{(i)}$

Ans:

$$
\begin{aligned}
&\min_{w,b,\xi} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} p^{(i)}\xi_i \\
&s.t.\ y^{(i)}(w^\intercal x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N, \quad \xi_i \geq 0
\end{aligned}
$$

ii. Derive the dual problem

Ans: Ans: $p^{(i)}$ adds an upper bound for the feasible set for the dual variables α_i

Introduce Lagrangian multipliers α_i and μ_i where $\alpha_i \geq 0$ and $\mu_i \geq 0$.

$$
L(w, b, \xi, \alpha, \mu) = \frac{1}{2} \mid\mid w \mid\mid_2^2 + C\sum_{i=1}^{N} p^{(i)}\xi_i + \sum_{i=1}^{N} \alpha_i(1 - \xi_i - y^{(i)}(w^\intercal x^{(i)} + b)) - \sum_{i=1}^{N} \mu_i \xi_i
$$

Partially differentiate Lagrangian by w, b, and ξ; then, set them to 0s for optimization.

$$
\begin{aligned}
&\frac{\partial L}{\partial w} = 0 \implies w = \sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)} \\
&\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{N} \alpha_i y^{(i)} = 0 \\
&\frac{\partial L}{\partial \xi} = 0 \implies C\sum_{i=1}^{N} p^{(i)} - \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \mu_i = 0
\end{aligned}
$$

Given $\alpha_i \geq 0$ and $\mu_i \geq 0$, we can derive the following from $\frac{\partial L}{\partial \xi} = 0$

$$
\begin{aligned}
&C\sum_{i=1}^{N} p^{(i)} = \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \mu_i \\
&0 \leq \alpha_i \leq Cp^{(i)}
\end{aligned}
$$

Plug the results of the partial differentiations back to the primal problem to derive the dual problem.

$$
\begin{aligned}
&\frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)}\right)^\intercal \left(\sum_{j=1}^{N} \alpha_i y^{(i)} x^{(i)}\right) + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i y^{(i)} \left(\sum_{j=1}^{N} \alpha_j y^{(j)} x^{(j)}\right) - \sum_{i=1}^{N} \alpha_i y^{(i)} b \\
&= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\intercal} x^{(j)} + \sum_{i=1}^{N} \alpha_i
\end{aligned}
$$

Derive the soft-margin SVM

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\intercal} x^{(j)}$$

$$s.t. \sum_{i=1}^{N} \alpha_i y^{(i)} = 0,\ 0 \leq \alpha_i \leq Cp^{(i)},\ i = 1, \ldots, N$$

iii. Given $p^{(1)} = 1, p^{(2)} = \frac{1}{2}, p^{(3)} = 0$, What are the feasible sets for $\alpha_1, \alpha_2, \alpha_3$?
Ans:

- $0 \leq \alpha_1 \leq 2$
- $0 \leq \alpha_2 \leq 1$
- $\alpha_3 = 0$

iv. Assume the optimization objective for the L_2 soft-margin SVM as given, derive the dual problem.
Ans:
Introduce Lagrangian multipliers α_i and μ_i where $\alpha_i \geq 0$ and $\mu_i \geq 0$.

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2} \mid\mid w \mid\mid_2^2 + \frac{C}{2} \sum_{i=1}^{N} p^{(i)} \xi_i^2 + \sum_{i=1}^{N} \alpha_i (1 - \xi_i - y^{(i)} (w^\intercal x^{(i)} + b)) - \sum_{i=1}^{N} \mu_i \xi_i$$

Partially differentiate Lagrangian by w, b, and ξ; then, set them to 0s for optimization.

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)}$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

$$\frac{\partial L}{\partial \xi} = 0 \implies \xi_i = \frac{\alpha_i + \mu_i}{Cp^{(i)}}$$

Plug the results of the partial differentiations back into the primal Lagrangian to derive the dual problem.

$$\begin{aligned} L(\alpha, \mu) &= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)} + \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \frac{(\alpha_i + \mu_i)^2}{2Cp^{(i)}} - \sum_{i=1}^{N} \frac{(\alpha_i + \mu_i)^2}{Cp^{(i)}} \\ &= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)} + \sum_{i=1}^{N} \alpha_i - \frac{1}{2C} \sum_{i=1}^{N} \frac{(\alpha_i + \mu_i)^2}{p^{(i)}} \end{aligned}$$

Maximizing $L(\alpha, \mu)$ with respect to $\mu_i \geq 0$ gives $\mu_i^* = 0$.

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)\top} x^{(j)} - \frac{1}{2C} \sum_{i=1}^{N} \frac{\alpha_i^2}{p^{(i)}} \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y^{(i)} = 0, \quad \alpha_i \geq 0,\ i = 1, \ldots, N. \end{aligned}$$

2. Implementing Support Vector Machine. (25 pt)

(a) Recall the dual problems of SVM in Problem 1. We define the domain $\mathcal{C} = [0, \infty]^N = \{\boldsymbol{\alpha} : \alpha_i \geq 0\}$ for a hard-margin SVM, and $\mathcal{C} = [0, C]^N = \{\boldsymbol{\alpha} : 0 \leq \alpha_i \leq C\}$ for a soft-margin SVM. We can solve this dual problem by projected gradient descent, which starts from some $\boldsymbol{\alpha}_0 \in \mathcal{C}$ (e.g., $\mathbf{0}$) and updates as follows:

$$\boldsymbol{\alpha}_{t+1} = \Pi_{\mathcal{C}} \left[\boldsymbol{\alpha}_t - \eta \nabla f(\boldsymbol{\alpha}_t) \right].$$

Here $\Pi_{\mathcal{C}}[\boldsymbol{\alpha}]$ is the *projection* of $\boldsymbol{\alpha}$ onto $\mathcal{C}$, defined as the closest point to $\boldsymbol{\alpha}$ in $\mathcal{C}$:

$$\Pi_{\mathcal{C}}[\boldsymbol{\alpha}] := \underset{\boldsymbol{\alpha}' \in \mathcal{C}}{\arg\min} \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|_2.$$

If $\mathcal{C}$ is convex, the projection is uniquely defined. With such information, in your **written submission, prove that**

$$\left(\Pi_{[0,\infty)^N}[\boldsymbol{\alpha}] \right)_i = \max\{\alpha_i, 0\},$$
$$\left(\Pi_{[0,C]^N}[\boldsymbol{\alpha}] \right)_i = \min\{\max\{0, \alpha_i\}, C\}.$$

(7 pt)

Hint: In this setting, since we have exactly the same domain for α_i in $\mathcal{C}$ for all is, each α_i can be considered independently. In this case, the minimization of $\|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|$ can also be considered independently for each i.

(b) Implement an `svm_solver()`, using projected gradient descent formulated as above. Initialize your $\boldsymbol{\alpha}$ to zeros. See the docstrings in `hw3.py` for details. **(12 pt)**

Remark: In this problem, you are allowed to use the `.backward()` function in PyTorch. However, then you may have to use in-place operations like `clamp_()`, otherwise the gradient information is destroyed.

Library routines: `torch.outer, torch.clamp, torch.Tensor.backward, torch.tensor.detach, with torch.no_grad():, torch.Tensor.requires_grad_, torch.tensor.grad.zero_, .`

(c) Implement an `svm_predictor()`, using an optimal dual solution, the training set, and the test set. See the docstrings in `hw3.py` for details. **(6 pt)**

Hint: Just in this subproblem, feel free to use iterations.

Remark 1: You don't need to convert the output of `svm_predictor()` to ± 1. Please just return the original output of SVM i.e., $\boldsymbol{w}^\top \boldsymbol{u} + b$ at the bottom of Lecture 11 (SVM I) Page 13.

Remark 2: In Lecture 12 (SVM II) Page 12, for support vector $\boldsymbol{x}^{(i)}$ with $\alpha_i > 0$ for hard-margin SVM, the formula

$$y^{(i)}(\boldsymbol{w}^\top \boldsymbol{x}^{(i)} + b) - 1 = 0$$

holds. In this way, you can compute b. Similarly, you could obtain b for soft-margin SVM.

Theoretically, any support vector $\boldsymbol{x}^{(i)}$ can obtain the same b. However, due to the precision limit, it might obtain different values of b in the implementation. So in order to deal this, in this problem, you are required to **use the support vector $\boldsymbol{x}^{(i)}$ with the minimum α_i among all support vectors** to compute b with the formula above. It's guaranteed that there will be exactly one i with minimum α_i (i.e. no ties) in the test cases on GradeScope.

Remark 3: Note that you don't need to know the value of C in soft-margin SVM (also not passed as a parameter), since C must be the maximum value of $\boldsymbol{\alpha}$, and will be larger than the suggested value.

(a) Prove that $\left(\prod_{[0,\infty)^N}[\alpha]\right)_i = \max\{\alpha_i, 0\}$ and that $\left(\prod_{[0,C]^N}[\alpha]\right)_i = \min\{\max\{\alpha_i, 0\}, C\}$

- $\left(\prod_{[0,\infty)^N}[\alpha]\right)_i = \max\{\alpha_i, 0\}$
  For each coordinate i:
  $$\alpha_i' = \underset{\alpha_i' \geq 0}{\arg\min}(\alpha_i' - \alpha_i)^2$$
  From this, we can derive the minimizer
  $$\alpha_i' = \begin{cases} \alpha_i & (\alpha_i \geq 0) \\ 0 & (\alpha_i < 0) \end{cases}$$
  Therefore, we can conclude that:
  $$\left(\prod_{[0,\infty)^N}[\alpha]\right)_i = \max\{\alpha_i, 0\}$$

- $\left(\prod_{[0,C]^N}[\alpha]\right)_i = \min\{\max\{\alpha_i, 0\}, C\}$
  For each coordinate i:
  $$\alpha_i' = \underset{C \geq \alpha_i' \geq 0}{\arg\min}(\alpha_i' - \alpha_i)^2$$
  From this, we can derive the minimizer
  $$\alpha_i' = \begin{cases} C & (\alpha_i \geq C) \\ \alpha_i & (C \geq \alpha_i \geq 0) \\ 0 & (0 \geq \alpha_i) \end{cases}$$
  Therefore, we can conclude that:
  $$\left(\prod_{[0,C]^N}[\alpha]\right)_i = \min\{\max\{\alpha_i, 0\}, C\}$$

3. Linear Regression and ERM. (25 pt)

(a) **Robustness of Linear Regression.** Consider a 1-dimensional linear regression problem with a dataset containing N data points $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $x^{(i)} \in \mathbb{R}^1$. The loss function is given by:

$$\ell(\mathbf{w}) = \sum_{i=1}^{N} (y^{(i)} - w_1^\top x^{(i)} - w_0)^2$$

where $\mathbf{w} = [w_1, w_0]^\top$, $w_1, w_0 \in \mathbb{R}^1$ are real numbers. Let's also fix $w_0 = 1$.

i. Given a dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{5} = \{(1,2),(2,3),(3,6),(4,7),(5,10)\}$, solve for w_1. **(2 pt)**

ii. Give this dataset an unreasonable outlier $\{(x^{(i)}, y^{(i)})\}_{i=1}^{6} = \{(1,2),(2,3),(3,6),(4,7),(5,10),(6,180)\}$, solve for w_1. **(2 pt)**

iii. Let's use L_1 norm for the loss function $\ell(\mathbf{w}) = \sum_{i=1}^{N} \|y^{(i)} - w_1^\top x^{(i)} - w_0\|_1$. Given the dataset with outlier $\{(x^{(i)}, y^{(i)})\}_{i=1}^{6} = \{(1,2),(2,3),(3,6),(4,7),(5,10),(6,180)\}$, solve for w_1. **(2 pt)**

(b) **Lasso Regression.** Given a dataset containing N data points $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$. Let $\boldsymbol{X}$ denote an $N \times d$ matrix where rows are training points, $\boldsymbol{y}$ denotes an $N \times 1$ vector corresponding output value:

$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{(1)\top} \\ \vdots \\ \boldsymbol{x}^{(N)\top} \end{bmatrix}$$

We assume the bias term w_0 to be 0.

In Lasso Regression, we want to reduce the complexity of $\boldsymbol{w}$ by shrinking less important feature coefficients to zero. Specifically, we want to find the optimal vector $\boldsymbol{w}^*$, such that:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_1,$$

where $\lambda > 0$. To make analysis easier, let's assume training data has this property:

$$\boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{I}$$

i. Show that under the assumption of the dataset, $\boldsymbol{w}_i^*$ is only related to $\boldsymbol{X}_{\cdot i}, \boldsymbol{y}$ and λ, where $\boldsymbol{X}_{\cdot i}$ is the ith column of $\boldsymbol{X}$. **(2 pt)**

ii. Assume that $\boldsymbol{w}_i^* > 0$, what is the value of $\boldsymbol{w}_i^*$ in this case? **(2 pt)**

iii. Assume that $\boldsymbol{w}_i^* < 0$, what is the value of $\boldsymbol{w}_i^*$ in this case? **(2 pt)**

iv. From (ii.) and (iii.), what is the condition for $\boldsymbol{w}_i^*$ to be zero? How can you interpret that condition? **(3 pt)**

(c) **Ridge Regression.** In this problem, we will derive the explicit solution to the Ridge Regression problem, which minimizes the mean squared error with a regularization term that penalizes the squared length of the coefficient vector. Specifically, we fix a regularization parameter $\lambda > 0$, and aim to solve the following optimization problem:

$$\boldsymbol{w}^*, w_0^* = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d, w_0 \in \mathbb{R}^1} \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(i)} - w_0 \right)^2 + \lambda \|\boldsymbol{w}\|_2^2.$$

To simplify the analysis, we assume that both the response variable y and the feature vector $\boldsymbol{x}$ are centered, meaning:

$$\sum_{i=1}^{N} y^{(i)} = 0, \quad \sum_{i=1}^{N} \boldsymbol{x}^{(i)} = 0.$$

i. **Warm-up: Ridge Regression for $d = 1$.**
In this case, $\boldsymbol{x}$ is a scalar (one-dimensional). Write out the explicit solution for $\boldsymbol{w}^*$ and w_0^* in this setting. **(5 pt)**

ii. **General case: Ridge Regression for $d > 1$.**
The multivariate version of the problem can be written in matrix form as follows:

$$\boldsymbol{w}^*, w_0^* = \underset{\boldsymbol{w}\in\mathbb{R}^d, w_0\in\mathbb{R}^1}{\arg\min} \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} - w_0\mathbf{1}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2,$$

where $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ is a matrix that rows are training points, $\boldsymbol{y} \in \mathbb{R}^N$ corresponds to values, and $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones. Derive the explicit solution for $\boldsymbol{w}^*$ and w_0^* in this setting. **(5 pt)**

(a) i. Given a dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^5 = \{(1,2),(2,3),(3,6),(4,7),(5,10)\}$, solve for w_1.
First, partially differentiate the classification rule by w_1
Ans: $w_1 = \frac{89}{55} \approx 1.6181818182$

$$\ell(w) = \sum_{i=1}^{N}(y^{(i)} - w_1^\intercal x^{(i)} - w_0)^2$$

$$\frac{\partial \ell}{\partial w_1} = -2\sum_{i=1}^{N}(y^{(i)} - w_1^\intercal x^{(i)} - 1)x^{(i)}$$

$$0 = -2\sum_{i=1}^{N}(y^{(i)} - w_1^\intercal x^{(i)} - 1)x^{(i)}$$

$$w_1 = \frac{\sum_{i=1}^{N}(y^{(i)} - 1)x^{(i)}}{\sum_{i=1}^{N} x^{(i)2}}$$

$$w_1 = \frac{89}{55} \approx 1.6181818182$$

ii. Give this dataset an unreasonable outlier $\{(x^{(i)}, y^{(i)})\}_{i=1}^6 = \{(1,2),(2,3),(3,6),(4,7),(5,10),(6,180)\}$, solve for w_1.
Ans: $w_1 = \frac{1{,}163}{91} \approx 12.7802197802$

(b) i. Show that under the assumption of the dataset, $\boldsymbol{w}_i^*$ is only related to $\boldsymbol{X}_{\cdot\boldsymbol{i}}, \boldsymbol{y}$ and λ, where $\boldsymbol{X}_{\cdot\boldsymbol{i}}$ is the ith column of $\boldsymbol{X}$.
Ans: $w_i^* = w_i^2 - 2(X_i^\intercal y)w_i + \lambda \mid w_i \mid$
Expand the original formula

$$w^* = y^\intercal y - 2y^\intercal Xw + w^\intercal X^\intercal Xw + \lambda \mid\mid x \mid\mid_1$$

Rewrite it in quadratic form and remove the terms that w_i is not dependent on

$$w^* = y^\intercal y + \sum_{i=1}^{N} X_i^\intercal y w_i + \lambda \mid w_i \mid$$

$$= \sum_{i=1}^{N} X_i^\intercal y w_i + \lambda \mid w_i \mid$$

Defines a minimization problem $\min_{w_i} g(w_i)$ for each w_i

$$\min_{w_i} g(w_i) = w_i^2 - 2(X_i^\intercal y)w_i + \lambda \mid w_i \mid = w_i^*$$

With the formula, we can conclude that w_i^* is only dependent on X_i, y, and λ

ii. Assume that $\boldsymbol{w}_i^* > 0$, what is the value of $\boldsymbol{w}_i^*$ in this case?
Ans: $w_i^* = X_i^\intercal y - \frac{\lambda}{2}$ (**where** $X_i^\intercal y > \frac{\lambda}{2}$)
Given $w_i^* > 0$

$$\min_{w_i} g(w_i) = w_i^2 - 2(X_i^\intercal y)w_i + \lambda \mid w_i \mid = 0 \quad \text{(Partially differentiate by } w_i)$$

$$2w_i^* - 2(X_i^\intercal y) + \lambda = 0$$

$$w_i^* = X_i^\intercal y - \frac{\lambda}{2} \quad (\text{where } X_i^\intercal y > \frac{\lambda}{2})$$

iii. Assume that $\boldsymbol{w}_i^* < 0$, what is the value of $\boldsymbol{w}_i^*$ in this case?
Ans: $w_i^* = X_i^\intercal y + \frac{\lambda}{2}$ **where** $X_i^\intercal y < \frac{\lambda}{2}$
Given $w_i^* < 0$

$$\min_{w_i} g(w_i) = w_i^2 - 2(X_i^\intercal y)w_i - \lambda \mid w_i \mid = 0 \quad \text{(Partially differentiate by } w_i)$$

$$2w_i^* - 2(X_i^\intercal y) - \lambda = 0$$

$$w_i^* = X_i^\intercal y + \frac{\lambda}{2} \quad (\text{where } X_i^\intercal y < -\frac{\lambda}{2})$$

iv. From (ii.) and (iii.), what is the condition for $\boldsymbol{w}_i^*$ to be zero? How can you interpret that condition?
Ans: $w_i^* = 0 \implies -\frac{\lambda}{2} \leq X_i^\intercal y \leq \frac{\lambda}{2}$
Given the solutions derived in (ii.) and (iii.), we can conclude that:

$$w_i^* > 0 \implies X_i^\intercal y > \frac{\lambda}{2}$$

$$w_i^* < 0 \implies X_i^\intercal y < -\frac{\lambda}{2}$$

$$\implies w_i^* = 0 \implies -\frac{\lambda}{2} \leq X_i^\intercal y \leq \frac{\lambda}{2}$$

(c) Write out the explicit solution for w^* and w_0^* in this scalar setting.
Ans: $w^* = \frac{\sum_{i=1}^N x^{(i)} y^{(i)}}{\sum_{i=1}^N (x^{(i)})^2 + N\lambda}$, $w_0^* = 0$

$$w^*, w_0^* = \arg\min_{w \in \mathbb{R}, w_0 \in \mathbb{R}} \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - wx^{(i)} - w_0\right)^2 + \lambda w^2.$$

Partially differentiate by w_0

$$\frac{\partial}{\partial w_0} = -\frac{2}{N} \sum_{i=1}^N \left(y^{(i)} - wx^{(i)} - w_0\right) = 0$$

$$0 = -\frac{2}{N} \left(\sum_{i=1}^N y^{(i)} - w \sum_{i=1}^N x^{(i)} - N w_0 \right)$$

$$\because \text{With centered data, } \sum_{i=1}^N y^{(i)} = 0 \text{ and } \sum_{i=1}^N x^{(i)} = 0$$

$$\therefore w_0^* = 0$$

Partially differentiate by w

$$
\begin{aligned}
\frac{\partial}{\partial w} &= -\frac{2}{N}\sum_{i=1}^{N} x^{(i)}\left(y^{(i)} - wx^{(i)} - w_0\right) + 2\lambda w = 0 \\
0 &= -\frac{2}{N}\left(\sum_{i=1}^{N} x^{(i)}y^{(i)} - w\sum_{i=1}^{N}(x^{(i)})^2\right) + 2\lambda w \\
&\Rightarrow \left(\frac{1}{N}\sum_{i=1}^{N}(x^{(i)})^2 + \lambda\right) w = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}y^{(i)} \\
&\therefore w^* = \frac{\sum_{i=1}^{N} x^{(i)}y^{(i)}}{\sum_{i=1}^{N}(x^{(i)})^2 + N\lambda}
\end{aligned}
$$

(d) Write out the explicit solution for w^* and w_0^* in this general-case setting.
Ans: $w^* = \left(\frac{1}{N}X^\intercal X + \lambda I\right)^{-1}\frac{1}{N}X^\intercal y$, $w_0^* = 0$

Partially differentiate by w_0

$$
\begin{aligned}
&\frac{\partial}{\partial w_0} = -\frac{2}{N}\mathbf{1}^\intercal(y - Xw - w_0\mathbf{1}) = 0 \\
&\Rightarrow \mathbf{1}^\intercal y - \mathbf{1}^\intercal Xw - Nw_0 = 0 \\
&\because \text{With centered data, } \mathbf{1}^\intercal y = 0 \text{ and } X = 0 \\
&\therefore w_0 = 0
\end{aligned}
$$

Partially differentiate by w

$$
\begin{aligned}
&\frac{\partial}{\partial w} = -\frac{2}{N}X^\intercal(y - Xw - w_0\mathbf{1}) + 2\lambda w = 0 \\
&\Rightarrow \left(\frac{1}{N}X^\intercal X + \lambda I\right) w = \frac{1}{N}X^\intercal(y - w_0\mathbf{1})
\end{aligned}
$$

Given centered data, $\mathbf{1}^\intercal y = 0$ and $\mathbf{1}^\intercal X = 0$, we have

$$
w_0^* = 0, \quad w^* = \left(\frac{1}{N}X^\intercal X + \lambda I\right)^{-1}\frac{1}{N}X^\intercal y.
$$

4. Implementing Linear Regression. (25 pt)

This assignment guides you through building, refining, and optimizing a **Linear Regression** pipeline: OLS, Ridge (L2), and Lasso via ISTA with a log-transform for skewed targets. Complete the `TODO` sections in the provided Python code. You can either use the Jupyter Notebook (`hw3_q4.ipynb`) or the Python file (`hw3_q4.py`). If you are not familiar with a particular term, please see code for more details.

Submission Requirements: If you use the Jupyter notebook, please convert it to `hw3_q4.py` and submit it to Gradescope (along with any helper file such as `hw3_utils.py` if you used one). Please ensure your code runs end-to-end.

(a) Data Preparation & OLS Baseline **(6 pt)**

i. **Dataset & Split**: Load the Ames Housing dataset from OpenML (`name="house_prices"`). Split into **train/val/test = 70/15/15** with a fixed `random_state`.

ii. **Preprocessing Pipeline**:

A. Identify numeric vs. categorical columns from the *training* dataframe.

B. Impute **numeric** with train **median** and **categorical** with train **mode** (apply the same stats to val/test).

C. Align val/test features to train by using z-scores.

iii. **OLS (Normal Equation)**: Implement ($\mathbf{w} * \text{OLS} = \text{pinv}(X^\top X)X^\top y$). Evaluate on the test set and report **MSE** and **RMSE**.

(b) Ridge Regression (L2 / MAP) **(6 pt)**

i. **Closed-Form with Unpenalized Bias**: Implement [$\mathbf{w}{*}\text{ridge} = (X^\top X{+}\lambda I)^{-1}X^\top y, \quad \text{with } I_{00} = 0$] so the bias term is not penalized.

(c) Lasso **(7 pt)**

i. **ISTA Update**: For the objective function $J(w) = \frac{1}{n}\|Xw - y\|_2^2 + \lambda\|w_{1:}\|_1$ (no penalty on bias), implement the gradient update: $w_0^{(k+1)} = w_0^{(k)} - \alpha \cdot \left(\frac{2}{n}X^\top(Xw^{(k)} - y)\right)_0$.

ii. Include a simple convergence check for a sufficient iteration budget (early stopping).

(d) Log-Transform **(6 pt)**

A **log transformation** is a common data preprocessing technique that replaces each value x in a dataset with its logarithm, $log(x)$. **Skewness** is a statistical measure of a distribution's asymmetry; a right-skewed distribution, for example, has a long tail of high-value outliers. In linear regression, applying a log transform to a skewed variable can make its distribution more symmetric, which helps to stabilize the variance of the errors (residuals) and better satisfy the model's core assumptions. We then need the Duan smearing estimator to correct for the systematic underestimation, or bias, that occurs when back-transforming predictions from the log scale into their original, linear scale.

i. **Motivation Plot**: On the **training** target, plot histograms of (`SalePrice`) and (log(1 + `SalePrice`)).

ii. **Log-Target Experiments**: Fit **Ridge** and **Lasso (ISTA)** with the ($\log(1 + y)$) target. Back-transform to dollars using **Duan's smearing**: compute the smearing factor ($s = \mathbb{E}[\exp(e)] = \frac{1}{n}\sum_{i=1}^{n}\exp(e_i)$) from log-residuals on train. Log-residuals are the model's errors calculated on the logarithmic scale, representing the difference between the actual log-transformed target values and the predicted log-transformed values. Then predict ($\widehat{y} = (e^{X_{\text{test}}w} - 1) \cdot \text{s}$).