CS480/680, Spring 2025

Assignment 1

Designer: Argyris Mouzakis; Instructor: Hongyang Zhang

Released: May 7; Due: June 4, noon

Instructions

- We do not accept hand-written submissions.
- This assignment is due by noon on June 4, 2025.
- For questions labelled as "**coding**", please follow the instructions provided and implement the required features. Unless otherwise specified, all implementations should be in *Python* using a *Jupyter Notebook*. Before submission, please make sure that your code can run without any errors. Also, be sure to save the output of each cell, as any missing output may not be graded.
- Please submit the following TWO files to LEARN:
  - A write-up in PDF format: the written answers to ALL questions, including the reported results and plots of coding questions, in a single PDF file.
  - An IPYNB file: your implementations for ALL coding questions. Please save the output of each cell, or your coding questions may NOT be graded.

Question 1 (Perceptron and SVM - 6 points) *In this question, we will run the perceptron algorithm by hand, and then compare its performance to that of SVMs. Consider the following dataset consisting of points in* $\mathbb{R}^2$ *with their labels:*

$$\begin{aligned}(x_1, y_1) &\coloneqq ((-1, 1), -1),\\ (x_2, y_2) &\coloneqq ((1, -1), -1),\\ (x_3, y_3) &\coloneqq ((2, 2), 1),\\ (x_4, y_4) &\coloneqq ((0, 3), 1).\end{aligned}$$

1. *(2 points). Run the perceptron algorithm for two epochs with initial weight vector* $w_0 \coloneqq (0, 0)$*, initial bias term* $b_0 \coloneqq 0$*, and learning rate* $\eta \coloneqq 1$*. Show your calculations in detail, and give a plot of the points and the line corresponding to the pair of* (w, b) *you get at the end of the last iteration. Calculate the margin.*

2. *(4 points). Calculate the SVM solution for the same dataset. Again, plot the points and the resulting decision boundary. What is the margin you get this time?*

Question 2 (Multiclass Perceptron - 10 **points)** *In this question, we will focus on* multiclass *classification. We will consider various different ways of generalizing the perceptron algorithm to the multiclass setting, and run the implementations on the MNIST dataset.*

1. *(2 points). The lectures covered the one vs. all and one vs. one reductions, which reduce multiclass classification to training multiple binary perceptrons. Implement either of the two reductions, and train it on the MNIST dataset. The training process should last* 10 *epochs. Report the final errors on the training and test sets.*

2. *(2 points). We will now work towards developing another multiclass generalization of the perceptron algorithm. To do that, we will first revisit the binary perceptron algorithm. Assume that we have a dataset of size* n*, consisting of labeled points* $(x_1, y_1), \ldots, (x_n, y_n)$*, where* $x_i \in \mathbb{R}^d$ $y_i \in \{\pm 1\}$ *for all* i*. Also, let* $\boldsymbol{x}_i \coloneqq (x_i, 1)$*. Consider the following objective function:*

$$\min_{\boldsymbol{w}} \sum_{i=1}^{n} \max\left\{0, -y_i \left\langle \boldsymbol{w}, \boldsymbol{x}_i \right\rangle\right\}. \tag{1}$$

   Assuming learning rate $\eta = 1$*, show that the update rule which the perceptron algorithm performs in each iteration is equivalent to choosing one term from the above sum, calculating its derivative* $\boldsymbol{g}$*, and then performing the* gradient update $\boldsymbol{w} \leftarrow \boldsymbol{w} - \boldsymbol{g}$*.*

   [Hint: You do not have to consider the case $\langle \boldsymbol{w}, \boldsymbol{x} \rangle = 0$*. You will end up with a derivative that has two different branches, depending on whether* $y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle > 0$ *or* $y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle < 0$*.]*

3. *(2 points). Assume now that we again have* n *labeled points* $(x_1, y_1), \ldots, (x_n, y_n)$*, where* $x_i \in \mathbb{R}^d$ $y_i \in \{1, \ldots, c\}$ *for all* i*, i.e., there are* c *classes. To attain as high accuracy as possible, we consider training a model that will use* c *perceptrons, with weight vectors* $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c \in \mathbb{R}^{d+1}$ *(For the training process, we formulate the following objective:*

$$\min_{w_1, \ldots, w_c} \sum_{i=1}^{n} \max_{k} \left\{\left\langle \boldsymbol{w}_k, \boldsymbol{x}_i \right\rangle - \left\langle \boldsymbol{w}_{y_i}, \boldsymbol{x}_i \right\rangle\right\}. \tag{2}$$

   Show that when $c = 2$*, this reduces to the binary perceptron problem in (1).*

   [Hint: Try to identify a weight vector $\boldsymbol{w}$ *using some transformation.]*

   How would you implement perceptron's update rule in the setting where there are two weight vectors $\boldsymbol{w}_1$ *and* $\boldsymbol{w}_2$*, instead of just one weight vector* $\boldsymbol{w}$*?*

4. *(4 points). Based on the analogy to the binary case made in the previous question, develop and implement a multiclass perceptron algorithm that directly solves (2). Run your implementation on the MNIST dataset for* 10 *epochs, and report the final errors on the training and test sets.*

   [Hint: we want to predict as follows: $\widehat{y} = \underset{k=1,\ldots,c}{\operatorname{argmax}} \langle \boldsymbol{w}_k, \boldsymbol{x} \rangle$*, i.e., the class* k *whose corresponding* $\boldsymbol{w}_k$ *maximizes the inner product. Explain your algorithm (e.g., through pseudo-code).]*

Question 3 (Robust Linear Regression and Lasso - 2 **points)** *Consider the linear regression problem:*

$$\min_{w \in \mathbb{R}^d} \|Xw - y\|_2 ,$$

where $X \in \mathbb{R}^{n \times d}$ *and* $y \in \mathbb{R}^n$ *(we omit the offset* b *for simplicity). Now, suppose we perturb each feature independently, and we are interested in solving the* robust *linear regression problem:*

$$\min_{w \in \mathbb{R}^d} \max_{\|z_j\|_2 \leq \lambda, \forall j} \|(X + Z)\, w - y\|_2 ,$$

where $Z \coloneqq [z_1, \ldots, z_d] \in \mathbb{R}^{n \times d}$ *is the perturbation matrix.*
Prove that robust linear regression is exactly equivalent to (square-root) Lasso:

$$\min_{w \in \mathbb{R}^d} \|Xw - y\|_2 + \lambda \|w\|_1 ,$$

where $\|w\|_1 = \sum_{i=1}^{d} |w_i|$.
Show all your calculations.
[Hint: Repeatedly use the self-duality *of the* ℓ_2*-norm, i.e.,* $\|w\|_2 = \max_{\|u\|_2 \leq 1} \langle w, u \rangle = \max_{\|u\|_2 = 1} \langle w, u \rangle$*.]*

Question 4 (Ridge Regression - 2 **points)** *The ridge regression problem is known as the linear regression that penalizes the* ℓ_2*-norm of the weights, i.e:*

$$\min_{W} \frac{1}{n} \|WX - Y\|_F^2 + \lambda \|W\|_F^2 .$$

Prove that the solution to this problem is $W = YX^\top \left(XX^\top + n\lambda \mathbb{I}\right)^{-1}$.