CSCI 145 Problem Set 5

September 27, 2025

Submission Instructions

Please upload *your* work by **11:59pm Tuesday September 30, 2025.**

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions independently.
- Your solutions to theory questions must be written legibly, or typeset in LaTeX or markdown. If you would like to use LaTeX, you can import the source of this document (available from the course webpage) to Overleaf.
- I recommend that you write your solutions to coding questions in a Jupyter notebook using Google Colab.
- You should submit your solutions as a **single PDF** via the assignment on Gradescope.

Grading: The point of the problem set is for *you* to learn. To this end, I hope to disincentivize the use of LLMs by **not** grading your work for correctness. Instead, you will grade your own work by comparing it to my solutions. This self-grade is due the Friday *after* the problem set is due, also on Gradescope.

Problem 1: Regularization

In this problem, we'll investigate the effect of regularization in optimization. Consider an optimization problem with d weights $\mathbf{w} \in \mathbb{R}^d$. Let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be a loss function, and $\lambda > 0$ is a regularization hyperparameter. Define the optimal weights

$$\mathbf{w}_\lambda^* = \arg\min_{\mathbf{w} \in \mathbb{R}^d} \mathcal{L}(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2. \tag{1}$$

Suppose that $\lambda_1 \geq \lambda_2$.

Part A: Regularization Term

Show that

$$\|\mathbf{w}_{\lambda_1}^*\|_2^2 \leq \|\mathbf{w}_{\lambda_2}^*\|_2^2. \tag{2}$$

Hint: Use the optimality conditions i.e., the optimal solution to the optimization problem must be better than every other solution. Plus some algebra.

Part B: Objective Term

Show that

$$\mathcal{L}(\mathbf{w}_{\lambda_1}^*) \geq \mathcal{L}(\mathbf{w}_{\lambda_2}^*). \tag{3}$$

Hint: Use the previous part, plus some algebra.

Part C: Empirical Check

Adapt your code from the logistic regression problem last week so that the loss function is binary cross entropy (with logits) plus a regularization term. Write a function to return the training loss and ℓ_2 norm of the weights for every epoch with hyperparameter λ.

Make one plot with the objective term (BCE) by iteration and one plot with the regularization term by iteration for some $\lambda_1 > \lambda_2$ of your choice. Does the theory we did in the prior parts hold? If not, what went wrong?

Problem 2: Soft Margin SVMs

In this problem, we will derive the dual formulation of the soft margin SVM, solve for it numerically on a dataset, and plot the resulting decision boundary.

Part A: Dual Derivation

Consider the soft margin SVM with tradeoff parameter $C > 0$ given by:

$$\operatorname*{arg\,min}_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \xi_i \quad \text{s.t.} \quad y^{(i)}\left(\langle \mathbf{w}, \mathbf{x}^{(i)}\rangle - b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i. \tag{4}$$

By (minimally) adapting the steps from class, show that the dual formulation is given by:

$$\operatorname*{arg\,max}_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{1}{2}\sum_{i,j=1}^n \alpha_i y^{(i)}\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)}\rangle y^{(j)}\alpha_j + \sum_{i=1}^n \alpha_i \quad \text{s.t.} \quad \sum_{i=1}^n y^{(i)}\alpha_i = 0, \quad 0 \leq \alpha_i \leq C \quad \forall i. \tag{5}$$

Hint: In addition to setting the gradient of the Lagrangian with respect to $\mathbf{w}$ and b equal to 0, do so for $\boldsymbol{\xi}$.

Part B: `cvxopt` Formulation

We will be using the python library `cvxopt` to solve the dual problem. However, we'll need to convert to their formulation first: In particular, we'll call `cvxopt.solvers.qp`$(\mathbf{P}, \mathbf{q}, \mathbf{G}, \mathbf{h}, \mathbf{A}, \mathbf{b})$ for the quadratic programming problem:

$$\operatorname*{arg\,min}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{P}\boldsymbol{\alpha} + \mathbf{q}^\top\boldsymbol{\alpha} \quad \text{s.t.} \quad \mathbf{G}\boldsymbol{\alpha} \leq \mathbf{h}, \quad \mathbf{A}\boldsymbol{\alpha} = \mathbf{b}. \tag{6}$$

Define $\mathbf{P}, \mathbf{q}, \mathbf{G}, \mathbf{h}, \mathbf{A}, \mathbf{b}$ so that Equation 6 and Equation 5 are equivalent.

Part C: Code!

Load a 2D binary classification dataset in python. I used:

```
X, y = sklearn.datasets.make_classification(
    n_features=2, n_redundant=0, n_informative=2,
    random_state=0, n_clusters_per_class=1
)
y = 2*y - 1 # Convert y to -1, 1
```

Compute $\mathbf{P}, \mathbf{q}, \mathbf{G}, \mathbf{h}, \mathbf{A}, \mathbf{b}$ in terms of your dataset (you'll have to wrap the `numpy` matrices in `cvxopt.matrix`). Then solve for $\boldsymbol{\alpha}$, and find the optimal line $\mathbf{w}^*$ and b^*. Finally plot the data and your line.

Hint: Generative AI is your friend for documentation, debugging, and examples!