Assignment 1

DSC 190/291: Learning Theory through Formal Proof and Proof Presentation

UCSD · Spring 2026

Released: Friday, April 3 · Due: Friday, April 10, 11:59 PM

Overview

This assignment has three parts:

Part A: Set up your individual repository.

Part B: A theory problem from Week 1.

Part C: Implement the Perceptron algorithm and verify its mistake bound experimentally.

You will also write a short report on how you used AI throughout the assignment.

AI policy. AI assistance is allowed and expected. You may use AI for coding, debugging, designing experiments, and writing. However, you must understand and stand behind everything you submit. The AI usage report (Part D) is a required component.

Submission. Submit a single PDF on Gradescope containing all your answers, plots, and discussion. Also push your code to your repository (see Part A) for reproducibility.

Part A: Repository Setup (10 points)

Create an individual Git repository for this course. This repo will hold all your assignment submissions for the quarter.

1. Create a **public** GitHub repository for this course.
2. Add a `README.md` with your name (or alias) and a one-line description.
3. Set up an AI workflow file in the root of your repository:
   - If you use **Claude Code**: create a `CLAUDE.md` with instructions and context for Claude about your project.
   - If you use **Codex** (or another agent): create an `AGENTS.md` with equivalent guidance.
   - You may include both if you use multiple tools.

Start simple — you will refine this file throughout the quarter as you learn what works.

4. Create a directory `hw1/` for this assignment.

5. Share the repository link with the instructor by submitting it on Canvas (or the method specified in class).

Part B: Theory Problem (40 points)

This problem starts from the threshold story in lecture and then pushes it one step further.

Throughout, let $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \{-1, +1\}$, and let

$$h_\theta(x) = \text{sign}(x - \theta),$$

where we use the convention $\text{sign}(z) = +1$ for $z \geq 0$ and $\text{sign}(z) = -1$ for $z < 0$.

We will study sequences with one extra structural assumption beyond what was explicitly analyzed in lecture.

Δ-Separated Threshold-Realizable Sequences. We say a sequence $\big((x_t, y_t)\big)_{t=1}^{T}$ is Δ-*separated threshold-realizable* if there exist $\theta^* \in [0, 1]$ and $\Delta > 0$ such that

$$y_t = h_{\theta^*}(x_t) \qquad \text{and} \qquad |x_t - \theta^*| \geq \Delta$$

for every round t.

1. **From continuous thresholds to a finite class.**

   Design a finite grid $G \subseteq [0, 1]$ whose size depends only on Δ, and consider the associated finite threshold class

   $$\mathcal{H}_G = \{h_\theta : \theta \in G\}.$$

   Prove that for every Δ-separated threshold-realizable sequence, there exists some $\widetilde{\theta} \in G$ such that

   $$h_{\widetilde{\theta}}(x_t) = y_t \qquad \text{for all } t = 1, \ldots, T.$$

   Then use the Halving theorem from lecture to derive a mistake bound of order

   $$O(\log(1/\Delta))$$

   for an explicit online learner.

   Your answer should clearly state:

   - your choice of grid G,
   - the size of G as a function of Δ,
   - the resulting mistake bound.

2. **A positive margin from separation.**

   View thresholds as linear predictors by choosing a feature map

   $$\phi : [0, 1] \to \mathbb{R}^d$$

   and a unit vector u^* depending on θ^*.

   Prove that every Δ-separated threshold-realizable sequence is linearly separable with margin at least $c\Delta$ for some absolute constant $c > 0$ under your representation, while

   $$\|\phi(x)\| \leq R \qquad \text{for all } x \in [0, 1]$$

   for some absolute constant R.

   Then use the Perceptron theorem from lecture to derive an explicit mistake bound of order

   $$O(1/\Delta^2).$$

   Your answer should clearly specify your chosen feature map, the margin lower bound, the norm bound, and the final mistake bound.

3. **Comparison and interpretation.**

   Explain why the continuous-threshold impossibility phenomenon from lecture does *not* contradict Part 1.

   Then compare the two mistake bounds you obtained in Parts 1 and 2. Why do they scale differently with Δ? What is each argument measuring about the problem?

4. **Optional strengthening: audit an AI proof.**

   An AI assistant gives the following argument:

   > Because every example is at least Δ away from the true threshold, continuous thresholds effectively form a class of size $O(1/\Delta)$. Therefore any online learner, including Perceptron, must make at most $O(\log(1/\Delta))$ mistakes on every Δ-separated threshold-realizable sequence.

   Identify at least **two** mathematical problems with this argument. Then write a corrected statement that is true and prove it.

Part C: Perceptron — Implementation and Experiments (35 points)

In lecture, we proved that the Perceptron makes at most R^2/γ^2 mistakes on data that is linearly separable with margin γ and bounded $\|x_t\| \leq R$.

Your task is to verify this experimentally. Implement the Perceptron algorithm (as described in lecture), design a data generation procedure, and run experiments that let you answer the following questions.

Questions to investigate

1. How do you generate data where you *know* the margin γ and the bound R? What choices does this require?
2. How does the number of mistakes M scale with $1/\gamma^2$? Plot this relationship.
3. Is the theoretical bound R^2/γ^2 tight, or does the Perceptron do better in practice?
4. The bound is independent of the dimension d. Is this what you observe? Design an experiment to test this.
5. What happens as $\gamma \to 0$? Connect your observations to the threshold counterexample from lecture.
6. If you used AI to help write your code, how did you verify that the implementation is correct? What tests or checks would catch a subtle bug in the update rule or data generator?

Use Python for your implementation. Include your code, plots, and a short discussion of your findings in `hw1/`.

Part D: AI Usage Report (15 points)

Write a short report (roughly half a page) describing how you used AI in this assignment. Address:

1. Which parts of the assignment did you use AI for? (e.g., writing code, debugging, designing experiments, explaining concepts, writing the discussion)
2. Give at least one concrete example of an AI suggestion you **accepted** and why.
3. Give at least one concrete example of an AI suggestion you **rejected** or **modified**, and why.
4. How did you verify that the code and results were correct?

Place this report in `hw1/` as a PDF or Markdown file.