DSC 190/291 · Assignment 1

UCSD · Spring 2026

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

Overview

This assignment has three parts:

1. **Part A:** Set up your individual repository.
2. **Part B:** A theory problem from Week 1.
3. **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 encouraged in this course. You may use AI to learn the material, explore proof structure, test examples, debug code or formalizations, and improve exposition. However, you are responsible for checking correctness and for standing behind every proof step, derivation, formalization, experiment, and explanation you submit. Use AI as a collaborator, not as an oracle: do not submit anything you cannot explain and verify. The AI usage report is a required component of the assignment.

Submission. Submit a single PDF on Gradescope containing your write-up, figures, and discussion. Also place any supporting artifacts for the assignment in your course repository under the appropriate assignment directory. This may include code, Lean files, notebooks, scripts, data, or other materials needed to inspect or reproduce your work. Your submission should make it clear how the repository artifacts relate to the write-up.

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 $((x_t, y_t))_{t=1}^T$ is Δ-*separated threshold-realizable* if there exist $\theta^* \in [0, 1]$ and $\Delta > 0$ such that

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

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 $\tilde{\theta} \in G$ such that

   $$h_{\tilde{\theta}}(x_t) = y_t \quad \text{for all } t = 1, ..., T.$$

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

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

   Your answer should clearly state:
   - your choice of grid G,
   - the size of G as a function of Δ,
   - the resulting mistake bound.

   Solution.

   Assume first that $0 < \Delta \leq 1$. Let

   $$m = \left\lceil \frac{1}{\Delta} \right\rceil$$

   and choose the uniform grid

   $$G = \left\{ \frac{j}{m} \;\middle|\; j = 0, 1, ..., m \right\}.$$

   Then $|G| = m + 1 = \lceil \frac{1}{\Delta} \rceil + 1 \leq \frac{1}{\Delta} + 2$.

   We claim that this grid contains a threshold that agrees with the unknown continuous threshold on every Δ-separated sequence. Let

$$\tilde{\theta} = \frac{\lceil m\theta^* \rceil}{m},$$

so $\tilde{\theta} \in G$ and

$$\theta^* \le \tilde{\theta} < \theta^* + \frac{1}{m} \le \theta^* + \Delta.$$

Fix any round t. If $y_t = +1$, then by the definition of h_{θ^*} and the separation assumption,

$$x_t \ge \theta^* + \Delta.$$

Since $\tilde{\theta} \le \theta^* + \Delta$, we have $x_t \ge \tilde{\theta}$, and therefore

$$h_{\tilde{\theta}}(x_t) = +1 = y_t.$$

If $y_t = -1$, then

$$x_t < \theta^*$$

and the separation assumption gives in fact

$$x_t \le \theta^* - \Delta.$$

Since $\tilde{\theta} \ge \theta^*$, we have $x_t < \tilde{\theta}$, and therefore

$$h_{\tilde{\theta}}(x_t) = -1 = y_t.$$

Thus every Δ-separated threshold-realizable sequence is realized by the finite class $\mathcal{H}_G = \{h_\theta : \theta \in G\}$. Running Halving on this finite class gives

$$M \le \lfloor \log_2 |\mathcal{H}_G| \rfloor \le \left\lfloor \log_2 \left(\left\lceil \frac{1}{\Delta} \right\rceil + 1 \right) \right\rfloor,$$

which is $O\big(\log(\frac{1}{\Delta})\big)$.

If $\Delta > 1$, no nonempty sequence in $[0,1]$ can be Δ-separated from a threshold in $[0,1]$; the empty case is trivial.

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

   View thresholds as linear predictors by choosing a feature map

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

   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

   $$\|\varphi(x)\| \le R \quad \text{for all } x \in [0,1]$$

   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.

Solution.

Use the augmented feature map

$$\varphi(x) = (x, 1) \in \mathbb{R}^2$$

and, for the realizing threshold θ^*, define

$$u^* = \frac{1, -\theta^*}{\sqrt{1 + (\theta^*)^2}}.$$

Then $\|u^*\| = 1$, and

$$\langle u^*, \varphi(x) \rangle = \frac{x - \theta^*}{\sqrt{1 + (\theta^*)^2}}.$$

Therefore

$$\text{sign}(\langle u^*, \varphi(x) \rangle) = h_{\theta^*}(x)$$

whenever $x \neq \theta^*$. In a Δ-separated sequence, no example is equal to θ^*.

For every round t,

$$y_t \langle u^*, \varphi(x_t) \rangle = |x_t - \theta^* \frac{|}{\sqrt{1 + (\theta^*)^2}}.$$

Since $\theta^* \in [0, 1]$, we have $\sqrt{1 + (\theta^*)^2} \leq \sqrt{2}$, so

$$y_t \langle u^*, \varphi(x_t) \rangle \geq \frac{\Delta}{\sqrt{2}}.$$

Thus the margin is at least $\gamma = \frac{\Delta}{\sqrt{2}}$, so one may take $c = \frac{1}{\sqrt{2}}$.

The feature norm is also uniformly bounded:

$$\|\varphi(x)\| = \sqrt{x^2 + 1} \leq \sqrt{2}$$

for all $x \in [0, 1]$. Thus $R = \sqrt{2}$.

By the Perceptron theorem from lecture,

$$M \leq \frac{R^2}{\gamma^2} \leq \frac{2}{\frac{\Delta^2}{2}} = \frac{4}{\Delta^2}.$$

This is an explicit $O\left(\frac{1}{\Delta^2}\right)$ 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?

Solution.

There is no contradiction because the lecture counterexample does not assume a fixed positive separation from the threshold. The adversary in that construction repeatedly places points closer and closer to the limiting threshold. Equivalently, along longer prefixes the minimum distance to the threshold can go to zero. Part 1 assumes a fixed $\Delta > 0$ for the whole sequence and builds a grid whose resolution depends on that Δ.

The two bounds come from different sufficient descriptions of the same separated threshold problem.

In Part 1, the separation assumption lets us replace the continuous threshold by one of only $\lceil \frac{1}{\Delta} \rceil + 1$ grid thresholds. Halving measures the logarithm of this finite effective class size, giving $O\big(\log(\frac{1}{\Delta})\big)$ mistakes.

In Part 2, the same sequence is represented as a linear-separation problem with margin at least $\frac{\Delta}{\sqrt{2}}$ and radius at most $\sqrt{2}$. The Perceptron theorem measures Euclidean geometry through $\frac{R^2}{\gamma^2}$, giving at most $\frac{4}{\Delta^2}$ mistakes.

Thus the Halving argument measures a one-dimensional discretization size, while the Perceptron argument uses a general margin theorem for linear predictors. The Perceptron bound is computationally simple and applies broadly to linear prediction, but in this special one-dimensional threshold setting it is not as sharp as the grid-plus-Halving bound.

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.

   Solution.

   There are several problems with the quoted argument.

   First, the continuous threshold class has not literally become a finite class of size $O\big(\frac{1}{\Delta}\big)$. What is true is a representation statement: for any fixed Δ-separated sequence realized by some continuous threshold, there exists a grid threshold from a grid of size $O\big(\frac{1}{\Delta}\big)$ that agrees with that sequence.

   Second, the conclusion says "any online learner." A finite realizing class gives a mistake guarantee for a particular learner, such as Halving over that finite class. It does not imply that every online learner has the same mistake bound. A learner that always predicts $+1$ can make a mistake on every round of a repeated negative example.

Third, the argument names Perceptron without using the Perceptron theorem. The Halving bound follows from a finite class. The Perceptron theorem instead gives a margin bound, which under the representation above is $\frac{4}{\Delta^2}$, not $O\big(\log\big(\frac{1}{\Delta}\big)\big)$.

A correct statement is the following. For every $\Delta \in (0, 1]$, let $m = \lceil \frac{1}{\Delta} \rceil$ and $G = \left\{ \frac{j}{m} \;\middle|\; j = 0, 1, ..., m \right\}$. Every Δ-separated threshold-realizable sequence is realized by some $h_{\tilde{\theta}}$ with $\tilde{\theta} \in G$. Therefore Halving run on $\mathcal{H}_G$ makes at most $\left\lfloor \log_2\big(\lceil \frac{1}{\Delta} \rceil + 1\big) \right\rfloor$ mistakes on every such sequence.

The proof is exactly the grid argument from Part 1: choose $\tilde{\theta} = \frac{\lceil m\theta^* \rceil}{m}$. Then $\theta^* \le \tilde{\theta} \le \theta^* + \Delta$. Points labeled $+1$ satisfy $x_t \ge \theta^* + \Delta \ge \tilde{\theta}$, and points labeled -1 satisfy $x_t \le \theta^* - \Delta < \tilde{\theta}$. Thus the grid threshold is consistent with the whole sequence. Since the resulting finite class has size $\lceil \frac{1}{\Delta} \rceil + 1$, the Halving theorem gives the stated bound.

Also, using $\varphi(x) = (x, 1)$ and $u^* = \frac{1, -\theta^*}{\sqrt{1 + (\theta^*)^2}}$, the Perceptron theorem gives the separate valid guarantee $M \le \frac{4}{\Delta^2}$.

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 $\|x_t\| \le R$.

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/`.

Solution.

A reasonable experiment uses a planted unit separator w^* and generates examples with certified margin. Fix $R = 1$. For each target margin $\gamma \in (0, 1)$, sample labels $y \in \{-1, +1\}$ and vectors z orthogonal to w^* with $\|z\| \leq \sqrt{1 - \gamma^2}$, then set

$$x = y\gamma w^* + z.$$

This guarantees $\|x\| \leq 1$ and $y\langle w^*, x\rangle = \gamma$.

Run the standard Perceptron update on sequences generated this way, repeat over several random seeds, and plot the number of mistakes M against $\frac{1}{\gamma^2}$ together with the theoretical line $\frac{1}{\gamma^2}$. To test dimension independence, keep γ and R fixed while varying d and sampling z in the orthogonal subspace. To study the small-margin regime, decrease γ toward zero and record the increase in mistakes and variance across trials.

The implementation should include checks that every generated example satisfies $\|x\| \leq R$ and $y\langle w^*, x\rangle \geq \gamma$, and simple tests that the update occurs exactly on mistakes. In random data, Perceptron will usually do better than the worst-case bound; the bound is a guarantee, not a prediction of typical performance.

Part D: AI Usage Report

(15 points)

Write a short report describing how you used AI in this assignment. Do not just list tools; explain what role AI played in your work and how you checked the result. Address:

1. Describe the parts of the assignment for which you used AI. For example: exploring examples, proposing conjectures, checking algebra, debugging code or formalizations, or improving exposition.
2. Describe concrete AI suggestions you accepted and explain why.
3. Describe concrete AI suggestions you rejected or substantially modified, and explain what was wrong, incomplete, or unhelpful about them.
4. Describe how you verified the correctness of what you submitted. Be specific about the relevant kind of work in this assignment: proof, derivation, code, experiment, or exposition.

AI workflow. Also describe concrete updates to your AI workflow that resulted from this assignment. This may include changes to `CLAUDE.md`, `AGENTS.md`, prompts, checklists, scripts, or skills. **Explain the 5 most recent changes you made to your AI workflow and why**.

If you did not use AI for some part of the assignment, say so explicitly.

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