DSC 190/291 · Assignment 6

UCSD · Spring 2026

Released: Monday, May 11 · Due: Monday, May 18, 11:59 PM

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: Boosting Sparse Linear Predictors and the ℓ_1 Margin

(45 points)

This problem continues the sparse-linear-model theme from Homeworks 4 and 5. Let $\varphi : \mathcal{X} \to \{-1,+1\}^d$ be a fixed feature map. For each $j \in [d]$ and $\sigma \in \{-1,+1\}$, define $b_{j,\sigma}(x) = \sigma\varphi_j(x)$, and let $\mathcal{B} = \{b_{j,\sigma} : j \in [d], \sigma \in \{-1,+1\}\}$. For a $\{-1,+1\}$-valued predictor b, its edge under a distribution Q is $\frac{1}{2} - L_Q(b) = \mathbb{E}_Q \frac{[yb(x)]}{2}$.

Assume $\mathcal{D}$ is realizable with margin by an s-sparse predictor: there is $w^\star \in \mathbb{R}^d$ with $\|w^\star\|_0 \le s$ and $y\langle w^\star, \varphi(x)\rangle \ge 1$ for every (x,y) in the support of $\mathcal{D}$. Equivalently, $\frac{w^\star}{\|w^\star\|_1}$ has ℓ_1-normalized margin at least $\frac{1}{\|w^\star\|_1}$. When a finite sample is used, write $S = ((x_1,y_1),...,(x_n,y_n))$.

You may use without proof that sparse linear classifiers have VC dimension $O\big(s\log(e\frac{d}{s})\big)$, so exact sparse ERM has realizable sample complexity $O\Big(\frac{s\log(e\frac{d}{s})+\log(\frac{1}{\delta})}{\varepsilon}\Big)$ up to logarithmic factors, but is computationally difficult when s is part of the input.

1. **(12 points) From sparse margin to a weak coordinate.**

   Assume additionally that $\|w^\star\|_\infty \le B$. For any distribution Q supported on examples satisfying the margin condition, prove that some $b \in \mathcal{B}$ has $L_Q(b) \le \frac{1}{2} - \frac{1}{2\|w^\star\|_1}$, and hence $L_Q(b) \le \frac{1}{2} - \frac{1}{2sB}$.

   Then describe an $O(nd)$ weighted ERM weak learner for $\mathcal{B}$ on weighted examples $(x_i, y_i, D_i)_{i=1}^n$. Hint: relate weighted error to the signed correlation $\sum_i D_i y_i b(x_i)$.

2. **(14 points) Boosting guarantee and comparison with sparse ERM.**

   Run AdaBoost over $\mathcal{B}$. At every round, the reweighted distribution is supported on the same margin-realizable sample, so the previous part applies with $\gamma = \frac{1}{2sB}$. Prove the exponential training-error bound, choose T so that the training error is zero, and then use the sparse-

linear VC bound, applied with the number of activated coordinates in place of s, to obtain a realizable generalization guarantee.

State the resulting sample complexity up to logarithmic factors, and compare it with exact sparse ERM. Your comparison should identify the statistical price paid by boosting, the computational advantage, and the role of B.

3. **(9 points) Why the coefficient bound matters.**

   For odd $s = 2m + 1$, set $d = s$. Construct a distribution supported on s labeled examples with $y = +1$ and $\varphi(x) \in \{-1, +1\}^s$ such that some $w^\star \in \mathbb{R}^s$ has margin 1, but every coordinate predictor has edge at most $2^{-\Omega(s)}$. Also show that the realizing vector in your construction satisfies $\|w^\star\|_\infty = 2^{\Omega(s)}$.

   Hint: index rows by $0, (1,+), (1,-), ..., (m,+), (m,-)$ with weights $q_0 = 1$ and $q_{r,+} = q_{r,-} = 2^{r-1}$. It suffices to build an invertible $s \times s$ sign matrix whose every column has q-weighted sum 1; try a base column with $(+,-)$ on every pair, columns that flip one pair, and carry columns with $(-,-)$ on earlier pairs, $(+,+)$ on one pair, and $(+,-)$ later. Prove realizability, compute the best coordinate edge, and explain why this rules out any polynomial-in-s AdaBoost guarantee based only on sparsity.

4. **(10 points) Experiment: sparse boosting versus a convex surrogate.**

   Implement AdaBoost over the coordinate class $\mathcal{B}$. Compare an easy sparse-margin distribution of your choice with the construction from the previous part. Report training error, exponential loss, observed edge, support size, and normalized margin over rounds. Compare AdaBoost with one convex surrogate method, for example logistic regression or hinge-loss minimization with an ℓ_1 constraint or penalty. Explain what the experiment illustrates about support sparsity, coefficient size, ℓ_1 margin, and computational tractability.

Part B: Agnostic Halfspace Hardness via Boosting (40 points)

Part A used boosting constructively; here boosting is a reduction tool.

Let $\mathcal{X}_d = \mathbb{R}^d$ and $\mathcal{Y} = \{-1, +1\}$. Let $\mathcal{H}_d$ be the class of affine halfspaces $h_{w,b}(x) = \text{sign}(\langle w, x\rangle + b)$, with $\text{sign}(z) = +1$ for $z \geq 0$. Let $\mathcal{I}_{d,k}$ be the class of intersections of k halfspaces: the output is $+1$ iff all k halfspaces output $+1$. For a size function $k = k(d)$, polynomial-size means $k(d) \leq d^c$ for some fixed constant c, and $k(d) = \omega(1)$ means $k(d) \to \infty$.

Use these black-box hardness facts: under standard uSVP hardness, intersections of d^r affine halfspaces are not efficiently PAC learnable in the realizable case, even improperly, for every fixed $r > 0$; under RSAT, the same is true for intersections of $k(d) = \omega(1)$ affine halfspaces.

1. **(10 points) A weak halfspace inside an intersection.**

   Prove: if $\mathcal{D}$ is realizable by some $g \in \mathcal{I}_{d,k}$, then some affine halfspace has error at most $\frac{1}{2} - \frac{1}{2k^2}$.

   Hint: split on $p = \mathbb{P}[y = +1]$. For small p, use the constant -1 halfspace; for large p, average over the k halfspaces defining the realizing intersection.

2. **(10 points) From an agnostic learner to a weak learner.**

Suppose, hypothetically, that affine halfspaces are efficiently properly agnostically PAC learnable. Use the previous lemma to construct a weak learner for $\mathcal{J}_{d,k}$ in the realizable case. Give γ such that the returned halfspace has error at most $\frac{1}{2} - \gamma$, and explain why the weak learner is polynomial-time when $k(d) \leq d^c$ for a fixed constant c.

3. **(12 points) Boosting the weak learner.**

   Use AdaBoost and the boosted-halfspace VC bound $\tilde{O}(Td)$ to obtain a realizable learner for $\mathcal{J}_{d,k}$. At every round, the weighted distribution is supported on examples still realized by the same intersection. State the sample and runtime dependence up to logarithmic factors.

4. **(8 points) Consequence for agnostic halfspaces.**

   Prove the implication: if affine halfspaces are efficiently properly agnostically PAC learnable, then intersections of $k(d) \leq d^c$ affine halfspaces are efficiently learnable in the realizable case, for every fixed c.

   Explain the contradiction with the black-box hardness facts by taking $k(d) = d^r$ under uSVP, or any polynomially bounded $k(d) = \omega(1)$ under RSAT. Conclude the implication for efficient proper agnostic PAC learning of halfspaces.

Part C: 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.