DSC 190/291 · Assignment 5

UCSD · Spring 2026

Released: Monday, May 4 · Due: Monday, May 11, 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: Sparse Linear Predictors: Statistical Sparsity versus Agnostic Hardness (45 points)

This problem continues the sparse-linear-model theme from Homework 4. There, you studied sparse halfspaces through VC dimension, structural risk minimization, validation, and brute-force search over supports. Here you will see what changes in the agnostic proper setting: the statistical story survives, but the empirical 0-1 optimization problem can become computationally hard when the sparsity level is part of the input.

For $d \in \mathbb{N}$ and $k \leq d$, define the class of k-sparse homogeneous linear classifiers

$$\mathcal{H}_{d,k} = \{h_w : x \to \text{sign}(\langle w, x \rangle) : w \in \mathbb{R}^d, \|w\|_0 \leq k\},$$

where throughout this problem

$$\text{sign}(t) = +1$$

if $t \geq 0$, and $\text{sign}(t) = -1$ otherwise.

For a labeled sample $S = ((x_1, y_1), ..., (x_m, y_m))$, define

$$\textsf{AGREEMENT}_{\mathcal{H}_{d,k}}(S, K) = 1$$

iff there exists $h_w \in \mathcal{H}_{d,k}$ that correctly labels at least K examples in S.

1. **(7 points) What Homework 4 already gives you.**

   Recall from Homework 4 that

   $$\text{VCdim}(\mathcal{H}_{d,k}) \leq O\left(k \log\left(e\frac{d}{k}\right)\right)$$

for $1 \le k \le \frac{d}{2}$. For larger k, the usual dense-halfspace bound gives $\mathrm{VCdim}\big(\mathcal{H}_{d,k}\big) \le O(d)$. Do not reprove this bound. Explain what it implies for agnostic sample complexity if exact empirical risk minimization over $\mathcal{H}_{d,k}$ were computationally available.

Also recall the brute-force realizable algorithm from Homework 4: enumerate supports $T \subseteq [d]$ with $|T| \le k$, and for each support solve a linear feasibility problem. State its runtime up to polynomial factors in the feasibility solver. In which regimes of k is this polynomial in d? Explain why this is a statistical/computational distinction rather than a contradiction.

2. **(24 points) Hardness of agreement via set cover.**

   Reduce SETCOVER to $\mathsf{AGREEMENT}_{\mathcal{H}_{d,k}}$. A SETCOVER instance consists of a universe $U = \{u_1, ..., u_r\}$, sets $A_1, ..., A_q \subseteq U$, and an integer $k \le q$. The question is whether there exists $R \subseteq [q]$ with $|R| \le k$ and $\bigcup_{j \in R} A_j = U$.

   Construct a labeled sample and threshold K such that the set-cover instance is a yes-instance iff there is a k-sparse homogeneous linear classifier that correctly labels at least K examples. The sample may be treated as a multiset, so repeated examples count with multiplicity.

   Hint: use one coordinate per set. Think of negative coordinates of w as the selected sets. Positive examples should charge the number of selected sets, while repeated negative examples should reward covered universe elements. You may use $M = q + 1$ repeated examples if helpful.

   Prove both directions of the reduction.

   Then invoke the Week 5 learner-to-agreement theorem. Explain in two or three sentences how it applies to the family $\mathcal{H}_{d,k}$ when both d and k are part of the input. Conclude that, if $\mathsf{NP} \neq \mathsf{RP}$, there is no efficient proper agnostic PAC learner for sparse linear classifiers when k is part of the input.

3. **(14 points) Experiment and interpretation.**

   Implement a small experiment comparing proper sparse 0-1 search with a convex surrogate method. Use synthetic data of your choice.

   Your experiment should include:

   - a proper sparse 0-1 search baseline for small d and k. For example, enumerate supports T with $|T| \le k$, then either enumerate a finite grid of weights on each support or use any exact small-sample method you can justify. State clearly whether your baseline is exact or approximate;
   - a convex surrogate method, for example logistic regression or hinge-loss minimization with an ℓ_1 constraint or penalty;
   - a comparison of runtime scaling as d or k increases;
   - a comparison of empirical 0-1 agreement and the surrogate objective or validation error.

   In your write-up, explain what the experiment illustrates about the difference between empirical 0-1 agreement over k-sparse classifiers and convex surrogate optimization over an ℓ_1-controlled linear predictor. Your explanation should also say whether the surrogate method returns a proper k-sparse classifier.

Part B: Convex Fixed-Feature Learning and Its Limits (40 points)

This problem studies the fixed-feature convex-learning template and its limits. First, you will see how hinge-loss minimization with an ℓ_1 constraint becomes a linear program. Second, you will show that optimizing a surrogate need not optimize the best 0-1 classifier. Third, you will prove the fixed-feature parity barrier stated in lecture. Finally, you will see why learning the feature map, as neural networks do, destroys convexity.

In Part B, the 0-1 margin loss counts zero margin as an error. This is separate from Part A's tie-breaking convention $\mathbf{sign}(0) = +1$ for sparse classifiers.

For binary labels $y \in \{-1, +1\}$, define the 0-1 loss of a real-valued score $f(x)$ as

$$\ell_{0\text{-}1}(f(x), y) = \mathbf{1}[yf(x) \le 0],$$

and define the hinge loss as

$$\ell_{\text{hinge}}(f(x), y) = (1 - yf(x))_+.$$

1. **(10 points) Hinge loss with an ℓ_1 constraint as a linear program.**

   A **linear program** is an optimization problem with a linear objective and linear equality or inequality constraints. For example, the variables might be $z_1, ..., z_N$, the objective might be $\min \sum_j c_j z_j$, and the constraints might have the form $\sum_j a_{ij} z_j \le b_i$. The word **linear** means that variables are only multiplied by constants and added together; expressions such as $z_i z_j$, z_i^2, $|z_i|$, and $(z_i)_+$ are not themselves linear. A common trick is to introduce extra variables and linear constraints whose optimum reproduces a non-linear-looking expression.

   Let $f_w(x) = \langle w, x \rangle$ with $w \in \mathbb{R}^d$. Fix a radius $B > 0$ and consider

   $$\min_{w:\ \|w\|_1 \le B} \frac{1}{m} \sum_{i=1}^{m} (1 - y_i \langle w, x_i \rangle)_+.$$

   Introduce variables $\xi_i \ge 0$ for the hinge losses and nonnegative variables w_j^+, w_j^- with $w_j = w_j^+ - w_j^-$. Write a linear program whose optimal value equals the optimization problem above. Explain the role of each family of constraints.

   Then explain why this linear program is not solving empirical 0-1 agreement over k-sparse classifiers.

2. **(10 points) A concrete counterexample: the surrogate comparator can be bad.**

   Fix $p \in \left(0, \frac{1}{2}\right)$ and $M > \frac{1-p}{p}$. Consider homogeneous one-dimensional linear predictors $f_w(x) = wx$ and the distribution $\mathcal{D}_{p,M}$ supported on two labeled examples:

   $$\mathbb{P}[(x, y) = (1, +1)] = 1 - p, \quad \mathbb{P}[(x, y) = (-M, +1)] = p.$$

   Compute the population hinge risk as a piecewise-linear function of w, and determine all of its minimizers. Then prove the following statements exactly:
   - $\inf_w L_{\mathcal{D}}^{0\text{-}1}(f_w) = p$;
   - every population hinge-risk minimizer has 0-1 risk equal to $1 - p$.

   Conclude that for any $\varepsilon > 0$ and any $\alpha < 1$, there are p, M such that

   $$\inf_w L_{\mathcal{D}}^{0\text{-}1}(f_w) \le \varepsilon$$

but every hinge-risk minimizer has 0-1 risk greater than α.

This example is intentionally low-dimensional. The difficulty is not algebraic; it is recognizing exactly what the surrogate objective is optimizing.

3. **(12 points) The fixed-feature parity barrier: proof and tightness.**

   Work on $\mathcal{X} = \{-1, +1\}^d$. For each subset $I \subseteq [d]$, define the parity function

   $$\chi_I(x) = \prod_{i \in I} x_i.$$

   In lecture, we stated that representing all parities with a fixed linear feature map requires exponentially many features. This problem proves that claim and shows that the bound is tight.

   Suppose a fixed feature map $\varphi : \{-1, +1\}^d \to \mathbb{R}^D$ can represent every parity as an exact linear predictor: for every $I \subseteq [d]$, there exists $w_I \in \mathbb{R}^D$ such that

   $$\forall x \in \{-1, +1\}^d, \quad \chi_I(x) = \langle w_I, \varphi(x) \rangle.$$

   This is exact equality as real-valued functions, not merely equality after applying sign.

   Prove that $D \geq 2^d$.

   Hint: form the $2^d \times 2^d$ matrix H whose rows are indexed by $I \subseteq [d]$, columns are indexed by $x \in \{-1, +1\}^d$, and entries are $H_{I,x} = \chi_I(x)$. Show that the rows of H are orthogonal, hence $\mathrm{rank}(H) = 2^d$. Then use the factorization $H = W\Phi$ induced by the feature map.

   Then prove the matching upper bound: define a feature map with one coordinate for each subset $I \subseteq [d]$, namely

   $$\varphi(x) = (\chi_I(x))_{I \subseteq [d]}.$$

   Show that every parity is represented exactly by a linear predictor in this feature space.

   Explain how the lower and upper bounds connect to the Week 5 message: fixed-feature convex learning can be statistically and computationally clean, but the required feature dimension may be exponential.

4. **(8 points) Same predictors, different optimization geometry.**

   Let $S = ((x_1, y_1), ..., (x_m, y_m))$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Consider squared loss.

   The fixed-feature linear model is $f_\beta(x) = \langle \beta, x \rangle$ with empirical risk

   $$L_{\mathrm{lin}}(\beta) = \frac{1}{m} \sum_{i=1}^{m} (\langle \beta, x_i \rangle - y_i)^2.$$

   The two-layer linear network with one hidden unit, with parameters $u \in \mathbb{R}^d$ and $v \in \mathbb{R}$, is

   $$f_{u,v}(x) = v\langle u, x \rangle$$

   with empirical risk

   $$L_{\mathrm{net}}(u, v) = \frac{1}{m} \sum_{i=1}^{m} (v\langle u, x_i \rangle - y_i)^2.$$

   Prove the following:

- the two parameterizations represent the same set of linear predictors;
- L_{lin} is convex in β;
- L_{net} is not necessarily convex in (u, v) by giving a concrete dataset S and a Jensen inequality violation;
- if β^* minimizes L_{lin}, then every factorization $\beta^* = vu$ gives a global minimizer of L_{net}.

Explain what this example shows about replacing fixed features by learned features.

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.

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

In addition to the standard report, answer the following two questions:

1. Which part of this assignment did AI help with the most: theorem proving, counterexample search, reduction design, coding, or exposition?
2. Describe one place where AI gave a plausible answer that was incomplete, ambiguous, or wrong, and explain the independent check that caught it.

Include this report in the same PDF.