Homework 1

Stats 315A, Winter 2026

January 15, 2026

Due date: Wednesday, January 21 at 11:59pm on Canvas.
Submission instructions: Upload three separate files to Canvas.

- A primary `.pdf` file of with your answers to the questions.
- A `.txt` with your predictions for problem 5
- A `.pdf` file with your code used to generate results for problems 1 and 5.

Question 1 (Linear vs. Knn, **10 points**): You are to run a simulation to compare KNN and linear regression in terms of their performance as a classifier, in the presence of increasing number of noise variables. You are welcome to use the programming language of your preference; if you do not have a strong prior preference, solve the problem using PyTorch in a Google Colab notebook. Submit your code (which should be relatively concise) as a separate PDF file.

(a) Implement the following setup, which builds on an example from Hastie et al. (2009, Chapter 2.3).

- Read in the 20×2 mean matrix `mixture_means.csv`. The first 10 rows are for class 1, the next 10 for class 2.
- Write a function to generate a sample of N points from a mixture of 10 Gaussians in $\mathbb{R}^2$, with each Gaussian density having scalar covariance with standard deviation `sigma` for each component. The function takes as inputs the `mean` matrix, `N`, and `sigma`, and outputs a matrix `X` of samples.
- Use your function to generate a training set of size 100 in each of the two classes, with $\texttt{sigma}^2 = 1/5$, as well as a test set of size 10K in each class. Create the corresponding response vectors for each. This should leave you with `xtrain`, `ytrain`, `xtest` and `ytest`.
- Write a function to compute the Bayes classifier for this setup. It should take as input the 20×2 matrix of means, `sigma`, and an input matrix `X`. Your function should classify all the rows of `X`.
- Write an evaluation function that takes as input your training and test data. In addition it should take as input a vector or list of values for k, the KNN neighborhood size parameter. Your function should
  - (i) Estimate the Bayes error using the test data.
  - (ii) Estimate the test error of a linear classifier fit by least squares (yes, one can use least squares for classification)
  - (iii) Estimate the test errors for KNN at each of the values of k.

(b) Run your function using $k = 1, 3, 5, 7, 9, 11, 13, 15$. Report your results for the Bayes predictor, linear classifier, and KNN for each k.

(c) Write a modification of your function used in the previous step that takes two extra parameters: the number `noise` of noise variables, and `sigma_noise`. The idea is to add additional Gaussian noise columns to `xtrain` and `xtest`. Your function should produce the list of test errors as before. Run your function with $1, 2, \ldots, 10$ noise variables, with `sigma_noise=1`. Summarize what you have learned.

Question 2 (CV for risk estimation and model selection, **10 points**): This question asks you to interpret and derive aspects of propositions 1 and 2 of Wager (2020). Your answers can be qualitative, and need not have have the rigor of a formal mathematical proof.

(a) In your own words, how does proposition 1 relate to a main claim of the note? Answer in no more than three sentences.

(b) In your own words, how does proposition 2 relate to a main claim of the note? Answer in no more than three sentences.

(c) Supposing assumption (1) of Wager (2020) holds exactly for the simulation set-up in the paper for some values $\gamma_{\text{random\_forest}} \neq \gamma_{\text{xg-boost}}$, how does proposition 2 predict figure two would change in the limit as $n \to \infty$. Consider both (i) the overlap in the green and purple distributions in the left panel and (ii) the fraction of points on each side of the dashed line in the right panel.

(d) Consider the decomposition

$$\widehat{\mathrm{CV}}_{n,K}(A) = \underbrace{\widehat{\mathrm{CV}}^{*}_{n,K}}_{\mathrm{I}} + \underbrace{2Z_{n,K}(A)}_{\mathrm{II}} + \underbrace{\Delta^2_{n,K}}_{\mathrm{III}}(A).$$

In the setting where n is very large and under assumption (1), which term(s) among I, II, and III will typically have the largest and smallest magnitude? How does your answer relate to proposition 1?

(e) Consider a similar decomposition of the difference in the CV estimates for two algorithms

$$\widehat{\mathrm{CV}}_{n,K}(A) - \widehat{\mathrm{CV}}_{n,K}(A') = (\mathrm{I} - \mathrm{I}') + (\mathrm{II} - \mathrm{II}') + (\mathrm{III} - \mathrm{III}')$$

In the same setting as part (e), which difference(s) will typically have the largest and smallest magnitude. How does your answer relate to proposition 2?

Question 3 (squared loss minimizers, **8 points**): Consider a k-class classification problem on a domain $\mathcal{X}$, where $f : \mathcal{X} \to \mathbb{R}^k$ is the prediction function and the loss $\ell : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}$ measures the loss of a prediction $f(x)$ on an example (x, y). Give the function minimizing the expected loss $L(f) := \mathbb{E}[\ell(f(X), Y)]$ for the following choices of loss $\ell : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}$.

(a) The squared error $\ell(v, y) = \frac{1}{2} \|v - e_y\|_2^2$, where e_y is the yth standard basis vector.

(b) The squared hinge loss

$$\ell(v, y) = (1 - v_y)_+^2 + \sum_{j \neq y} (1 + v_j)_+^2 ,$$

where $(t)_+ = \max\{t, 0\}$. *Hint.* If $p^\star = [P(Y = y \mid X = x)]_{y=1}^k$ is the vector of conditional probabilities of Y given X, you might use that $1 - p^\star_y = \sum_{j \neq y} p^\star_j$.

Question 4 (Ridge regression risks, **10pts**): Consider the ℓ_2-regularized (ridge) regression estimator

$$\widehat{\beta}_\lambda := \operatorname*{argmin}_b \left\{ \frac{1}{2} \|Xb - y\|_2^2 + \frac{\lambda}{2} \|b\|_2^2 \right\},$$

where $X = [x_1 \ \cdots \ x_n]^T \in \mathbb{R}^{n \times p}$ is the (fixed) design matrix and $y \in \mathbb{R}^n$ is the response. Let $H_\lambda = X(X^T X + \lambda I)^{-1} X^T$, and assume that

$$y_i = f(x_i) + \varepsilon_i \tag{0.1}$$

where $\mathbb{E}[\varepsilon_i] = 0$ and $\mathbb{E}[\varepsilon_i^2] = \sigma^2$. (Note that we *do not* assume that $y_i = x^T \beta^\star + \varepsilon_i$.) Recall also that the in-sample risk of an estimate $\widehat{f}$ of f is

$$R_{\text{in}}(\widehat{f}) := \frac{1}{n} \sum_{i=1}^n \mathbb{E}[(\widehat{f}(x_i) - f(x_i))^2] = \frac{1}{n} \sum_{i=1}^n \left(\text{Bias}(\widehat{f}(x_i))^2 + \text{Var}(\widehat{f}(x_i)) \right)$$

where the expectation is taken over y_i drawn in model (0.1). Define the mean values of y by

$$\mu := \mathbb{E}[y] = [f(x_i)]_{i=1}^n.$$

Throughout the remainder of the question, let $\widehat{f}_\lambda$ be the linear function $\widehat{f}_\lambda(x) = x^T \widehat{\beta}_\lambda$ given by the ridge estimator.

(a) Show that

$$n \cdot R_{\text{in}}(\widehat{f}_\lambda) = \|(I - H_\lambda)\mu\|_2^2 + \sigma^2 \operatorname{tr}(H_\lambda^T H_\lambda).$$

(b) Show that the residual sum of squares $\text{RSS} = \sum_{i=1}^n (\widehat{f}_\lambda(x_i) - y_i)^2$ (this is just the training squared error) satisfies

$$\mathbb{E}[\text{RSS}] = n R_{\text{in}}(\widehat{f}_\lambda) + \sigma^2 (n - 2 \operatorname{tr}(H_\lambda)).$$

For the remainder of the question, assume that the design $X \in \mathbb{R}^{n \times p}$ has rank p, that is, it is full column rank.

(c) Let $X = U \Gamma V^T$ be the singular value decomposition (SVD) of X, where $U \in \mathbb{R}^{n \times p}$ satisfies $U^T U = I_p$ and $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_p)$ is the diagonal matrix of singular values. Using this SVD, give as explicit a formula as you can for the derivative matrix

$$\dot{H}_\lambda := \frac{\partial}{\partial \lambda} H_\lambda \in \mathbb{R}^{n \times n}.$$

(d) Let $r(\lambda) = n \cdot R_{\text{in}}(\widehat{f}_\lambda)$ be the in-sample risk as a function of $\lambda \geq 0$. Give a formula for the derivative $r'(\lambda) = \frac{\partial}{\partial \lambda} r(\lambda)$.

(e) Using your preceding two answers, show that $r'(0) < 0$, that is, there is *always* some $\lambda > 0$ so that the in-sample risk of the ridge estimator is smaller than unregularized least squares.

Question 5 (A loan data analysis challenge, **10 points**): A company in Chile uses crowdsourcing to fund loans to the public, as a means to offer relief from the high bank interest rates. The data in this challenge consists of historical loan records for a sample of 9000 past customers. The variables characterize some aspects of the loan, such as duration, amount, interest rate and many other more technical features of the loans. There are also a number of qualitative variables, such as reason for

loan, quality rating of the borrower and others. The response variable y of interest is `default`: a 0-1 variable indicating whether or not the borrower has defaulted on their loan payments.

The company would like to build a default risk score so that they can target high-risk customers early and perhaps preempt the default event, which ends up costly for all involved. (The fraction of defaults in the entire population is around 7%.) The training data `loan-train.csv` represents the sample 9000 past customers, and contains 30 features and the binary outcome `default` (in the first column). The file `loan-testx.csv` consists of a random sample of 10000 other customers from the general pool. For these you are provided only the 30 features.

Your job is to build a *risk score*, that is, a model that estimates the probability of default $y = 1$. Feel free to use any of the tools discussed in the lectures of this class (or beyond). Some packages that may be useful include `pytorch`, `scikit-learn`, `xgboost`, and just regular old linear regression.

(a) Describe your approach to the problem and any key assumptions that underlie it. The evaluation for this part does not depend on the accuracy of your predictor.

(b) Produce a risk score for each of the 10000 other customers and submit a text file with 10000 lines; on each line, your should have your predicted risk estimate for each test customer, in the same order as `loan-testx.csv`. You will be evaluated on the accuracy of your predictor.

(c) Suppose that the training and test samples were not iid, but that you could make the following assumptions:

- The training set and test set samples are iid from a population with known proportions of defaulting customers, α_{Train} and α_{Test}, respectively.
- The customers in the training set who default (and who do not default) are representative of those in the test population who default (and who do not default).
- You are able to fit from training set a predictor $f_{\text{Train}} : x \to [0, 1]$ such that for each x, $f_{\text{Train}}(x)$ is the probability that a sample from the training population defaults.

Provide an expression for a Bayes optimal predictor under zero-one loss for whether a customer from the test population defaults using α_{Train}, α_{Test} and f_{Train}.

Question 6 (Time and collaboration, **2 points**)**:**

(a) How many hours in total did you spend on this assignment? (This will help to calibrate future assignments.)

(b) With whom (if anyone) did you collaborate on this assignment?

References

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, 2 edition, 2009.

Stefan Wager. Cross-validation, risk estimation, and model selection: Comment on a paper by rosset and tibshirani. *Journal of the American Statistical Association*, 2020.