Final Exam

CSCI 1420—Spring 2024

Instructions

Timeline: The final will be posted on the course homepage no later than noon Eastern U.S. time on Thursday, May 16. It must be submitted on Gradescope by 11:59 PM Eastern U.S. time on Friday, May 17. No late days may be used on the final. For every minute past the deadline it is late, one percentage point will be deducted from the grade.

Exam Format: There are 5 problems, each worth 1/5 of the exam. The intention is that the questions increase in difficulty, but this is, of course, only approximate.

Submission Format: You may submit a PDF using the provided Latex, or you may submit handwritten answers. You are encouraged to use Latex if possible, as any illegible parts of answers will be marked incorrect.

Academic Integrity: *The course collaboration policy does not apply to this exam.* Instead, you may not communicate with anyone other than course staff about the exam in any way. You may consult the course textbook, course notes, slides, homeworks, recorded lectures, or existing messages on Ed. You may not post anything new that is public to Ed during the exam period. (See below.) Violating these instructions will be considered academic dishonesty.

Getting Help: If you have any clarification questions about the content of the exam or technical difficulties, please first consult the "Official FAQ" that will be pinned on Ed. If your question is not answered there, please email the HTA list: `cs1420headtas@lists.brown.edu` . We will respond as soon as possible, but please keep in mind that latency of up to 20 minutes is reasonable. In addition, the mailing list is only monitored from 9 AM to 10 PM Eastern U.S. time, so please plan accordingly.

If you have any other issues or concerns, such as challenging or unexpected circumstances, please contact Steve directly: `stephen_bach@brown.edu` .

Problem 1 (The Bias-Complexity Tradeoff)

Consider a binary ($\mathcal{Y} = \{-1, 1\}$) classification task with inputs in $\mathcal{X} = \{0, 1\}^d$, where d is large (> 1000). Rank the following four hypothesis classes in this setting according to the bias-complexity tradeoff, starting with the most bias (i.e., least complexity). Indicate your rank with a number beside each hypothesis class (e.g., 1 is the most bias and 4 is the least bias).

After each choice, briefly (1–2 sentences) justify your answer. You may cite without proof any facts contained in lectures, homeworks, or the textbook.

Reminder: the difference between homogeneous and non-homogeneous halfspaces is that non-homogeneous halfspaces have decision boundaries that are *not* constrained to pass through the origin. In other words, they contain "bias" parameters.

_____ Decision trees

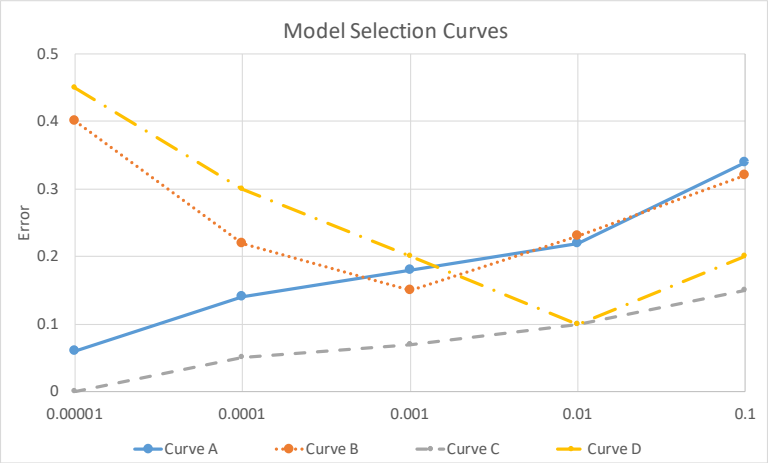
_____ Non-homogeneous halfspaces

_____ Homogeneous halfspaces

_____ Boosted, non-homogeneous halfspaces with an ensemble size of 2

Problem 2 (Model Selection)

Consider these (partially labeled) model selection curves for two ℓ_2-regularized logistic regression classifiers:



The training of the two classifiers differed only in the input data. The first model ("Model 1") was trained by regularized risk minimization using the log loss. The second model ("Model 2") was trained in an identical fashion, but had some features *removed* from each example.

Describe the likely interpretation of the following parts of the above figure, based on the bias-complexity tradeoff. Include a specific statement of what that part of the figure represents, and provide a brief explanation justifying your interpretation.

(a) The horizontal axis (with values 10^{-5} through 10^{-1}):

(b) Curve A (solid line):

(c) Curve B (dotted line):

(d) Curve C (dashed line):

(e) Curve D (dashed/dotted line):

Problem 3 (VC Dimension)

Consider the hypothesis class from lectures of intersections of halfspaces. Specifically, let $\mathcal{H}$ be the set of hypotheses defined by the intersection of two-dimensional, homogeneous halfspaces. Each hypothesis in $\mathcal{H}$ is defined by two weight vectors $\mathbf{w}_1 \in \mathbb{R}^2$ and $\mathbf{w}_2 \in \mathbb{R}^2$, where:

$$h_{\mathbf{w}_1,\mathbf{w}_2}(\mathbf{x}) = \begin{cases} 1 & \text{if } \langle \mathbf{w}_1, \mathbf{x} \rangle > 0 \text{ and } \langle \mathbf{w}_2, \mathbf{x} \rangle > 0 \\ -1 & \text{otherwise} \end{cases}$$

and $\mathbf{x} \in \mathbb{R}^2$ and $\mathcal{Y} = \{-1, 1\}$.

What is the VC dimension of this hypothesis class $\mathcal{H}$? Provide a complete proof.

Problem 4 (Polynomial Kernels)

In homework 9, problem 1 we proved the polynomial kernel is a valid kernel. For this question we will assume we are performing binary classification of points $x \in \mathbb{R}$, $\mathcal{Y} = \{-1, 1\}$. Take the expression of the polynomial kernel in problem 1, and set $b = 1$ and $c = 1$ such that we get $K(\mathbf{u}, \mathbf{v}) = \langle \psi(\mathbf{u}), \psi(\mathbf{v}) \rangle = (1 + \langle \mathbf{u}, \mathbf{v} \rangle)^d$.

(a) Find the function $\psi(x) : \mathbb{R} \to \mathbb{R}^{d+1}$ for the polynomial kernel K above (with $b = 1$ and $c = 1$). In other words, find the function that maps a point x into the corresponding higher dimensional feature space.
Hint: By the Binomial Theorem, $(a + b)^n = \sum_{i=0}^{i=n} \binom{n}{i} a^{n-i} b^i$.

(b) Given the following three points in $\mathbb{R}$ and associated labels in {-1,1}:

x	y
1	1
3	-1
5	1

Give one possible equation for a parabola of the form $f(x) = c(x - a)(x - b)$ such that the points with label 1 are above the curve and the points with label -1 are below it. In other words, find a, b, and c such that $x > f(x)$ if x has a label of 1 and $x < f(x)$ otherwise. *Hint:* Plot the points along a number line and draw a parabola.

(c) Show that an SVM using this kernel can perfectly classify any set with n examples where $n \leq d + 1$. You can describe the polynomial that separates the data or an algorithm for constructing it.
Hint: Think about how part (b) might generalize.

Problem 5 (XGBoost)

`XGBoost` ("eXtreme Gradient Boosting") is a very popular regularized *gradient tree boosting* library. It has become particularly famous for its use in many winning submissions in the data science competitions, such as on Kaggle. It turns out that the `XGBoost` algorithm is fairly similar to algorithms we saw this semester, albeit with some extensions. A few are highlighted below.

For this question, you may consult the `XGBoost` documentation linked above and the original paper here.

Residuals

In a nutshell, `XGBoost` is an ensemble of regression trees, i.e., decision trees that perform continuous regression. A key difference between `XGBoost` and `AdaBoost` is that, in `XGBoost`, each ensemble member does not make a prediction (e.g., a label) directly. Instead it produces a *residual*, which is not a value to be returned but a value to be added to the ensemble's total. That total is the prediction of the model. It could represent the answer to a continuous regression problem or a score for a class in a classification problem. At each leaf of each regression tree, there is a scalar weight w that the tree outputs if the input example reaches that leaf.

(a) What is a neural network architecture (either one we discussed in class or not) that uses residuals to produce its output?

Regularization

`XGBoost` finds an ensemble h that (approximately) minimizes the following regularized risk:

$$L_S(h) = \sum_{i=1}^{m} \ell(y_i, h(\mathbf{x}_i)) + \sum_{k=1}^{K} \Omega(f_k), \qquad \Omega(f_k) := \gamma T_k + \frac{1}{2}\lambda \|\mathbf{w}_k\|_2^2 \tag{1}$$

over $1 \leq k \leq K$ regression trees f_k where ℓ is the chosen loss function. T_k is the number of leaves in the tree f_k and $\gamma, \lambda \geq 0$ are regularization coefficients. $\mathbf{w}_k \in \mathbb{R}^{T_k}$ is the vector of leaf weights in the tree.

(b) What do the regularization coefficients control? What should we expect when increasing them?

(c) A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if, for all $t \in [0, 1]$,

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).$$

Supposing f and g are convex functions, prove that $f + g$ is convex.

(d) When is Equation (1) convex? Why is convexity desirable?

Fast Computation

`XGBoost` is an *additive model* in the sense that the overall model h is updated with tree f_t according to $h_t = h_{t-1} + f_t$. At step t, to select the split for the new tree f_t to add to the ensemble, we want to minimize

the following:.

$$L_S^{(t)}(f_t) = \sum_{i=1}^{m} \ell(y_i, h_{t-1}(\mathbf{x}_i) + f_t(\mathbf{x}_i)) + \Omega(f_t). \tag{2}$$

(e) Write out the 2nd order Taylor expansion of $L_S^{(t)}$ in (2) with respect to $h_{t-1}(\mathbf{x}_i)$. You can express your answer in terms of partial derivatives of $\ell(y, h_{t-1}(\mathbf{x}_i))$.

(f) How many terms/factors are constant with respect to f_t, i.e., only need to be computed once per iteration? Compare the original objective against the 2nd order approximation. How do they differ regarding what is constant?

(*Reminder:* f_t is varied as different splits are evaluated.)