Statistical Learning, Fall 2024-5

Example final No. 2

1. (40 points) **Minimum norm interpolation**

In this problem we investigate the behavior of regression models in high dimension, show how we can define the "best" interpolating solution via a choice of regularization and investigate the empirical implications of this result with standard linear regression and kernel methods. Assume as usual that we have n observations and p covariates (+intercept=$p+1$ total), and assume $p \geq n$. Define as usual the design matrix $\mathbb{X}$ as the matrix with p columns and n rows.

(a) Assume $\mathbb{X}$ is full rank, and state explicitly what its rank is in that case. Explain why it implies that the standard linear regression problem will result in interpolation, i.e.:

$$\min_{\beta_0,\beta} \sum_i (y_i - x_i^\intercal \beta - \beta_0)^2 = 0$$

i. Does this result uniquely define the solution?

ii. Will the conclusion be different if we use a linear model with absolute loss? Quantile loss?

(b) Now assume we add a little ridge penalty regularization to our linear regression problem, which becomes:

$$\min_{\beta_0,\beta} \sum_{i=1}^{n} (y_i - x_i^\intercal \beta - \beta_0)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Denote the solution to this problem by $\hat{\beta}(\lambda)$. Argue that this (unique!) optimal solution will no longer be an interpolating one.

Hint: One way to do this is by looking at the derivatives with regard to β_j at any interpolation point and arguing that they are different from zero.

(c) Define $\hat{\beta}^{(l2)}$, the minimal-2-norm interpolator, as the interpolation solution which minimizes the ridge penalty:

$$\min \sum_{j=1}^{p} \beta_j^2 \quad \text{subject to} \quad \sum_i (y_i - x_i^\intercal \beta - \beta_0)^2 = 0$$

When $\lambda > 0$, argue that this solution has the smallest ridge-penalized loss of any interpolating solution.

(d) Given a non-interpolating model $\tilde{\beta}$, show that there is $\epsilon > 0$ such that for $\lambda < \epsilon$ the solution $\hat{\beta}^{(l2)}$ has smaller penalized loss than $\tilde{\beta}$.

(e) Combine the previous results to show that as $\lambda \to 0$, the solution $\hat{\beta}(\lambda)$ converges to $\hat{\beta}^{(l2)}$.

Note: A formal proof is appreciated but clear relevant arguments will also be accepted.

(f) (* Extra credit) Derive an explicit algebraic expression for the minimum Euclidean norm interpolator $\hat{\beta}^{(l2)}$.

(g) Extending the result from part (d) (in this case, short non-formal correct arguments are enough, no need for formal proofs):

i. What would be the equivalent result if we used lasso regularization instead of ridge regularization?

ii. How would the result change if we used absolute or quantile loss instead of squared error loss?

2. **Short problems — 8 points each**

(a) In a regression problem, we are given $n = 200$ training observations in $p = 1000$ dimensions, and also a very large test set from the same distribution. We are told that when performing PC-regression on $d = 1$ principal components in this data, we get the following results:

- The first PC captures over 80% of the overall variance of the training data x vectors
- Performing 1-dimensional PC regression (i.e., linear regression with only this PC) does extremely well in prediction

Is Ridge regression likely to work well in this setting? Explain.

(b) We take a data set, divide it into equally-sized training and test sets, and then try out two different classification procedures. First we use logistic regression and get a misclassification error rate of 20% on the training data and 30% on the test data. Next we use 1-nearest neighbors (i.e. K = 1) and get an average error rate (averaged over both test and training data sets) of 18%. Based on these results, which method should we prefer to use for classification of new observations?

(c) Assume we build a Neural Network for regression as described in class, with one or more hidden layers, with one or more nodes in each hidden layer, and nonlinear transformation σ. For example, a model with two hidden layers, and two nodes in each, and p explanatory variables, will be (as we showed in class):

$$f(x) = \sum_{i=1}^{2} w_i^{(3)} \sigma \left(\sum_{j=1}^{2} w_{ji}^{(2)} \sigma (\sum_{k=1}^{p} w_{kj}^{(1)} x_k) \right).$$

Assume that for the "nonlinearity" function σ we decide to use the absolute value function $\sigma(u) = |u|$. What can we say about the resulting model? Is it a reasonable choice compared to standard non-linearities like sigmoid or relu? Or does it suffer from some issues that rule it out as a non-linearity choice?

3. **Longer problems — 12 points each**

(a) You have some training data $(\mathbb{X}, \mathbb{Y})$ and are considering fitting two simplistic models to it:

$$\begin{array}{ll} \text{Model 1:} & \hat{Y} = \bar{y} \text{ the average of the training data} \\ \text{Model 2:} & \hat{Y} = y_1 \text{ the first observation.} \end{array}$$

(Both models ignore $\mathbb{X}$ completely, and for both the prediction is identical for all observations). Now consider the expected fixed-X squared prediction error of these models. As we discussed, this error can be decomposed into expected training error + optimism, which is defined as:

$$optimism = \frac{2}{n} \sum_i cov(y_i, \hat{y}_i).$$

Denote the expected training error by t_1 for model 1 and t_2 for model 2, their respective optimism by o_1, o_2 and their respective expected prediction errors by p_1, p_2. For each of t, o, p explain whether the value for model 1 is bigger, equal or smaller than for model 2, or whether it's impossible to say.

(b) Given a training set, assume we apply K-NN with k neighbors using leave-one-out-CV (n-fold CV) to the data and calculate the sum of squared errors, denote it $RSS_k(LOO)$. We then apply regular K-NN with $k+1$ neighbors to the same data (no cross validation), and calculate the training sum of squared errors, denote it $RSS_{k+1}(Train)$. Calculate:

$$\frac{RSS_{k+1}(Train)}{RSS_k(LOO)}.$$

Hint: Write the prediction of a specific observation in each one of the two scenarios and find the relationship between the two expressions.

(c) We described gradient boosting as taking the (negative) derivative of the loss after $t-1$ iterations and using that as the response for finding the weak learner in iteration t. We than explained the use of residual as (half) the negative derivative of squared loss $L(y, \hat{y}) = (y-\hat{y})^2$. Using the notation from class, where $F^{(t-1)}$ is the model after $t-1$ iterations, we can denote this:

$$y_i^{(t)} = y_i - F^{(t-1)}(x_i).$$

Now assume instead of taking half the negative derivative, we take the derivative itself:

$$y_i^{(t)} = 2\left(y_i - F^{(t-1)}(x_i)\right).$$

We compare two boosting algorithms run on the same data: the first (version A) uses the first version with the residual, and the second (version B) with twice the residual. Both algorithms use CART trees of a fixed depth (say 2) as weak learners. We choose a value of ϵ_A and number of iterations T_A and build a boosting model $F_A^{(T_A)}$ using version A. Are there values ϵ_B, T_B that will give the exact same model on the same data ($F_B^{(T_B)} = F_A^{(T_A)}$ when running version B instead? Explain.