STAT 2270: Data Mining
University of Pittsburgh
Fall 2020

Homework 1

1. Consider the generic regression setup

$$Y = f(X) + \epsilon$$

where ϵ is the irreducible error, which we'll assume is independent of X and is i.i.d. with mean 0 and variance σ^2. Suppose that we observe a dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$ from which we construct an estimator $\hat{f}$ for the true regression function f. Now consider a new test point $X = x$.

(a) Show that the expected squared error loss (risk) of $\hat{f}$ at x can be decomposed into a term measuring the difference between $\hat{f}$ and f plus another term involving the irreducible error. Given this, what's the "best" test error we could hope for with an estimated regression function?

(b) Show further that the term from part (a) involving $\hat{f}$ and f can be further decomposed into the bias and variance of $\hat{f}$. Use this to write the expected squared error loss of $\hat{f}$ at x as the sum of three familiar terms.

2. In class, we discussed the intuition behind why we need to constrain the kind of regression function we're using to model the data. Here we'll make this formal. Suppose we have a finite dataset $\mathcal{D}_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ consisting of a single response $Y \in \mathbb{R}$ and a single predictor $X \in \mathbb{R}$.

(a) Show that for any finite sample, there will always exist an infinite (uncountable) number of regression function estimates with perfect fit (zero loss) with respect to L_p loss for any p.

(b) Show that in fact this holds for even an infinite sample $\{(x_1, y_1), (x_2, y_2), ...\}$.

(c) Given a generic regression setup of the form $Y = f(X) + \epsilon$, a regression estimate $\hat{f}$ is said to be an *interpolator* if it fits the training data perfectly (i.e. $\hat{f}(x) = y$ for all $(x, y) \in \mathcal{D}_n$). Sticking to the one-feature context, show that for any finite dataset there always exists a near-optimal interpolator under squared error loss. That is, for any $\epsilon > 0$, show that there always exists some interpolator $\hat{f}$ such that the distance between f and $\hat{f}$ is less than ϵ.

We'll talk about the idea of interpolation much more towards the end of the class. There's a lot more to be said but for now, you should take from part (c) that it's not "surprising" that a regression estimate could interpolate the training data and still be "good".

3. Suppose that we observe a dataset of the form $\{(x_1, y_1), ..., (x_n, y_n)\}$ consisting of a response $Y \in \mathbb{R}$ and predictors X and we choose to use a linear model

$$Y = \beta^T X + \epsilon$$

to estimate the relationship. However, suppose that in reality the true model for Y is given by

$$Y = \beta^T X + g(Z) + \epsilon^*$$

where Z denotes an additional collection of features that we don't observe. When is our linear model approximation appropriate? *Hint: Consider what form the error will take. When are the standard linear model assumptions and conditions violated?*

4. Here we'll make use of the `Hitters` dataset found in the `ISLR` package in R. Note that ISLR here stands for the textbook "**I**ntroduction to **S**tatistical **L**earning with applications in **R**" cited in the syllabus for this class. You may find the lab contained in Chapter 6 of ISLR very useful for the first few parts of this exercise.

(a) Load in the `Hitters` dataset found in the `ISLR` package in R. Following the approach in ISLR, we'll use `Salary` as our response and treat the rest of the variables as predictors. Remove all rows of the data that contain missing values. We'll treat the remaining data as our working dataset for the remainder of this problem.

(b) Construct a linear model and record the resulting MSE of the predictions on the (training) data. Don't worry about fixing up the linear model (e.g. removing terms that are not significant) – simply regress `Salary` on all remaining predictors and record the error.

(c) Now construct the best possible linear model you can find. You can do this however you like: removing terms that are not significant, using combinations of forward/backward selection, including interaction terms, including higher order polynomial terms etc., but don't use any of the regularization methods we discussed in class. Once you've found what you think is the best model, record the MSE.

(d) Find the best model using ridge regression. Consider a range of tuning parameters and select the best model by cross validation. Record the value of the optimal tuning parameter as well as the values of the resulting coefficient estimates from that model.

(e) Repeat part (d) with lasso.

(f) Repeat part (d) with elastic net.

(g) Compare the models in parts (b)-(f). Does any one model (or a few models) stand out as being substantially better than the others?

(h) Make a scatterplot of coefficient estimates from part (b) vs those found in part (d). Repeat this to compare the results in (b) with those from (e) and also from (f). Are those coefficients with the smallest estimates those most penalized in parts (d)-(f)?

(i) Repeat parts (d)-(f) with a dataset that includes interactions and higher order polynomial terms. Do these models appear to be better than the others you've fit thus far?

(j) Now add some noise to the response. That is, replace `Salary` by `Salary` $+\epsilon$ where $\epsilon \sim N(0, \sigma^2)$ for some chosen value of σ. Again, try a standard linear model, ridge, lasso,

and elastic net where each model is optimally tuned. Repeat this for several increasing values of σ^2. On one plot, plot the error of each model against the amount of additional noise. Does one model or a certain subset of models tend to perform better or worse for large or small values of σ? Is there some intuition for this? Discuss.