LASSO and Ridge Questions from Class Stat 495

Do we need to know the l_1 and l_2 formulas?

Yes, I would make sure you understand the structure of the shrinkage penalties for both settings. (And which is which, etc.)

When should we use LASSO over ridge? Vice versa. Either of these over OLS. When should we NOT use LASSO or ridge?

OLS is more easily understandable by a general audience, or someone not that familiar with statistics. Inference in OLS is better understood. Ridge and LASSO tend to be for audiences with higher statistical knowledge.

If you think some of your variables should have coefficients near 0, use Ridge/LASSO. If you want variables eliminated in combination with that, use LASSO.

If $p > n$, you should use LASSO.

Tibshirani's paper also summarizes the performance of the methods in terms of number and size of effects. You can review that yourselves.

Not sure? Run all three (or both if you've eliminated OLS) and compare their performance on a test set. This is what your homework has you doing.

Is there a compromise method between LASSO and Ridge?

Yes, elastic-net models. For the glmnet function, if you set an alpha between 0 and 1, you'll be fitting an elastic-net model. The value of alpha you set determines if it's more ridge than LASSO or vice versa.

How do we interpret LASSO solutions?

This question could either be about the solution in terms of coefficients or the Lasso path pictures. I'll try to answer both. For coefficients, they are just the estimated slope parameters, so that's interpreted like OLS slopes. You just get an added benefit of variable selection with it. For understanding the path, like the figure, you can think of it as showing many different possible solutions, for different levels of constraint (a lot to none, or reversed, depending on how it is plotted). When you take a vertical slice in the plot, that's one possible solution, and you get the coefficients that correspond to that solution. Again, many different values can be used on the x-axis. The y-axis usually represents the Beta_hat values

(slope coefs). For the x-axis, remember that lambda = 0 is the OLS solution, and lambda = infinity pushes all coefs to 0. Use that to orient yourself.

How does multicollinearity affect these models?

In my experience, about the same as you'd expect with OLS models – one variable can do the "main" work of a group of highly correlated variables. However, there's less inferential theory developed for Ridge and LASSO, so I can't describe how badly variance estimates and test statistics might be distorted. You could run an OLS, check for multicollinearity through VIFs, and then use that to set the predictors you want to use in Ridge / LASSO, if you wanted to be particularly careful about it.

The one item I'd note there is these methods are typically used for prediction, so you may not "care" about multicollinearity, because you won't be trying to interpret which predictors are actually useful – you just want good predictions. Similarly, there are usually a large number of predictors, so trying to pre-check for this might be more time consuming than just running LASSO and letting it do variable selection.

Please explain how Figure 2 (question 4) demonstrates that LASSO results in coefficients of 0 while Ridge does not.

This figure shows the shape of the imposed constraints. For LASSO, constraining the absolute values of the coefficients in 2-D would yield this square (the sum of their absolute values is <= a constant). For Ridge, you end up drawing a circle in 2-D because you are saying the first slope squared + second slope squared must be <= a constant. The solution must satisfy the constraint, so the figure helps you imagine moving out from the OLS estimates to find a solution that meets the constraint (imagine the contours show solutions with similar performance). Eventually, you will reach the constraint. Where you hit the constraint determines the resulting coefficients.

For LASSO, the shape of the constraint has corners. This means a variable can have its slope set to 0. For Ridge, the corners don't exist – you'd get coefficients set very close to 0, but not at 0. (There is a tiny chance you'd hit a corner). In higher dimensions, the idea is the LASSO constraint has MANY corners, while Ridge doesn't. So, in LASSO, we expect coefficients to be set to 0.

In the later Figure, what is s/s_hat and how is it related to lambda from the CASI notes?

In Tibshirani's paper, he describes the constraint as setting the sum of the absolute values of the coefficients to be less than or equal to a constant **t**. (This is one way of writing it without lambdas). He defines **s** for the figure to be **t** over the sum of the absolute values of the OLS coefficients. So, if **s** is 1, then t = the sum of the abs(OLS) coefficients (theoretical betas). He is using **s** to describe the amount of

shrinkage. For the s_hat of 0.44 (the hat is there because it is what was estimated using the data), it's saying that 0.44 = t/sum(abs(estimated OLS coefs)). I.E. the constrained sum of the new beta estimates was 0.44 of the sum (abs(original OLS coefs)). So, it's describing the amount of shrinkage in relation to the OLS estimates. The larger s is, the larger lambda would be (as in, there's no constraint). So that's why you see coefficients at 0 when s_hat=0 (t=0), and getting larger as s_hat increases, until you get the OLS estimates back at s_hat = 1 (no shrinkage or selection). Again, there can be multiple different values used on the x-axis in the path diagrams. Try to orient yourself to where lambda is 0 and infinity based on where you see the OLS estimates and everything at 0, respectively.

How is the shrinkage penalty determined? How do we find the optimal lambda? Is it via grid search?

Yes, in combination with cross-validation, at least if you are using cv.glmnet to do it. See the help file:

https://glmnet.stanford.edu/reference/cv.glmnet.html

You can either set the lambda sequence or let glmnet do this (the default). Again, the default grid is usually fine.

What is the bias-variance tradeoff in terms of ridge and LASSO?

Ridge and LASSO both add bias to the slope coefficient estimates, and often, have an associated decrease in variation, such that when you compute the MSE, they improve on the OLS estimates. So both of these are examples where we intentionally added bias in order to get better estimates through a variance reduction.

How is LASSO computationally inexpensive? What are the computational advantages/disadvantages of LASSO / ridge?

This is a great question! There is a very nice section on why the LASSO is computationally inexpensive in CASI. The handout you have with note highlights gives part of the answer. The LASSO solution is piecewise linear, and we can find the knots, so it's not too hard to solve. There's also the question of the LASSO and degrees of freedom (short section in CASI). Basically, LASSO is like OLS – when a new variable enters the solution, a df is spent. This is better than forward selection (it searches a bigger space when finding solutions, sort of). As you saw in lab, the same function in R can be used to fit both Ridge and LASSO solutions, so Ridge isn't too computationally expensive either.

Could we get more information about how the equations for ridge and LASSO work in depth?

I'm not sure if this question is asking about their theoretical setup or how the solutions are found. CASI has a section on the LASSO math, but when I have tried to teach that before in 495, it was extremely challenging for students, so I wanted us to focus more on the principles and implementation. If we had a Stat computing class, it would be a good topic.

In terms of the math equations, I think the easiest way to think about it is that both start out like OLS. You want to minimize the residual sum of squares (RSS). However, both ridge and LASSO add penalties to the RSS to alter the final slope estimates you get. They use different penalties and both do shrinkage of the betas, but only LASSO does selection (can set slope coefficients to 0). In this way, they aim to improve on the OLS estimates. They both add bias to the estimates, but the reduction in variance can offer improvement over OLS.

Do we need to know much about the orthonormal design case?

There was a subsection in Tibshirani 1996 that discussed this. No, I wouldn't spend time on this, if you understand the general properties of the LASSO.

A few of you asked for more resources on particular implementations of the LASSO or other versions of the LASSO (such as logistic regression via the LASSO).

All I can say is there a lot of literature out there about these topics, and you can investigate that way. Potentially, you could use one of these as your topic for the final paper for the course.