Handout 04: Penalized Regression (Optional)

On handout 2, we considered a regression problem with an estimated value for β and a data matrix X factorized using the SVD as UDV^t. Then, we considered the predictions from a new $\tilde{\beta}$ equal to β plus a multiple of the smallest right singular vector (V_p). This is given by:

$$X(\tilde{\beta}) = X(\beta + aV_p) \tag{1.1}$$
$$= X\beta + aXV_p \tag{1.2}$$
$$= X\beta + a\sigma_p. \tag{1.3}$$

In the lab questions, you assumed $\sigma_p = 0$ and this shows that the predictions $\widehat{y}$ for β are exactly equivalent to the predictions for $\tilde{\beta}$. What if σ_p is positive but small? In this case the predictions are not exactly the same but they are still very difficult to distinguish. Under sufficient noise it is still nearly impossible to distinguish between these two solutions when σ_p is small. This can make regression very difficult to perform because large datasets often have a smallest singular value that is quite small (more on this later).

The fundamental problem here is that we are only the mean squared error as our loss function. Therefore, there is no easy way to distinguish between using β and $\tilde{\beta}$. One solution is to modify the loss function to make it easier to distinguish between these two solutions. For example, here is the equation for ridge regression:

$$\beta_\lambda = \arg\min_b \left\{ ||y - Xb||_2^2 + \lambda ||b||_2^2 \right\} \tag{1.4}$$

For some constant $\lambda > 0$. It says that you want to minimize the errors in prediction, but with an additional cost associated with large values of β. This helps to distinguish between the many possible models and often does a much better job than the ordinary least squares estimator at predicting future values. You can derive a very elegant solution to the ridge regression, particular when you incorporate the SVD. I will have you derive this in the following lab questions.

PCA

There is a closely related concept to ridge regression known as principal component analysis. It also comes quite cleanly from the SVD decomposition, but is not directly associated with predictive modelling. The question it attempts to answer is: How can we visualize a matrix X when p is large? The idea is to capture the variation in X in a small number of dimensions; usually this is done in two dimensions so that we can easily plot the dataset.

Consider the SVD of a matrix X:

$$X = UDV^t. \tag{1.5}$$

The most important terms, in terms of the overall size of the map X, are those corresponding with the largest singular values. To approximate this, we can remove all but the largest k singular values from d. Furthermore, in visualization, the final rotation V^t

Lasso Regression

Above, we looked at adding a penalty term to our loss function to prefer smaller regression vectors over larger ones. Adding an ℓ_2-penalty leads to the ridge regression, which has some nice properties. For example, we can write down an analytic expression for the form of the regression vector and could prove (though we did not) that it does an ideal job of minimizing the variance of estimated regression vector.

Today we will look at two other penalties that could be added to the sum of squared residuals. The first is called the ℓ_0-norm, though it is not in fact a vector norm. It counts the number of non-zero terms in a vector:

$$||b||_0 = \#\{j \quad \text{s.t.} \quad b_j \neq 0\}. \tag{1.6}$$

Adding this to the least squares estimator leads to best subset regression:

$$\beta_\lambda^{BSR} = \arg\min_b \left\{||y - Xb||_2^2 + \lambda||b||_0\right\} \tag{1.7}$$

As another alternative, we can use the ℓ_1-norm, given by the sum of absolute values of the coordinates:

$$||b||_1 = \sum_j |b_j|. \tag{1.8}$$

This is a proper vector norm. Adding it to the square errors leads to the lasso regression vector:

$$\beta_\lambda^{LASSO} = \arg\min_b \left\{||y - Xb||_2^2 + 2\lambda||b||_1\right\} \tag{1.9}$$

Best subset regression is useful when you have only a small number of variables. For large datasets it is computationally intractable because the optimization problem is not convex. The only way to find a solution is to check every single combination of variables; the number of possibilities explodes beyond just a few variables. The lasso regression does not have an analytic solution but can be approximated using iterative methods; it is a convex optimization task. What makes it so attractive is that it will do a form of subset selection that, in practice, is nearly as good as the best subset selection.

Deriving the iterative solutions for the lasso regression problem is fairly extensive and not applicable to many other applications. We will not get into the details in this course. Today you are going to work with the simple case where the columns of X are uncorrelated:

$$X^t X = 1_p. \tag{1.10}$$

In this particular example it is possible to find analytic solutions to both best subset selection and the lasso regression. I think it yields a lot of motivation for understanding the behavior of the lasso in the more general case.