Lecture 3

IDS575: Statistical Models and Methods
Theja Tulabandhula

Notes derived from the book titled "Elements of Statistical Learning [2nd edition] (Sections 3.2-3.4)

1 Bias Variance Tradeoff

Let us now understand the expressiveness or complexity of models. As seen in the previous section, such complexity is controlled by:

- regularizer coefficient
- kernel parameters, or
- number and type of basis functions.

We cannot use RSS(f), which is defined on training data, to determine these parameters. This is because we will always pick those that give the least residuals. Such models will fail spectacularly on test data.

Lets look at what the impact of the parameter is by first writing down the EPE at a test point x_0, and then specializing it for the k-nearest-neighbor method.

Let $Y = f(X) + \epsilon$ with $E[\epsilon] = 0$ and $Var(\epsilon) = \sigma^2$. Let x_i be non-random.

$$\begin{aligned} EPE(x_0) &= E_{\epsilon,\tau}[(Y - \widehat{f}(x_0))|X = x_0] \\ &= \sigma^2 + \text{Bias}^2(\widehat{f}(x_0)) + \text{Var}_\tau(\widehat{f}(x_0)) \end{aligned}$$

There are three key terms above:

- First term: Irreducible error σ^2. This is the variance of the new test output variable, and cannot be removed/reduced even if you know $f(x_0)$.
- Second term: Is called the bias term. it is the difference between the true mean $f(x_0)$ and the expected value of the estimate.

- Third term: is the variance of the estimate.

Example 1. The k-nearest-neighbor method: Bias is $f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_{(l)})$ and Variance is $\frac{\sigma^2}{k}$. Here $x_{(l)}$ is the l^{th} nearest neighbor. Bias may increase with increasing k because neighbors are further away. The variance is just the variance of the average, so it decreases as k increases. Thus, there is a bias-variance tradeoff with respect to k.

When model complexity increases, there is more variance and less bias. When model complexity decreases, there is more bias and less variance.

We want to choose model complexity to trade bias with variance so as to minimize test error (e.g, EPE). Training error (e.g., RSS) is not a good estimate of test error. Figure 1 shows an illustration of the behavior of test and training errors as model complexity of some model family is varied.

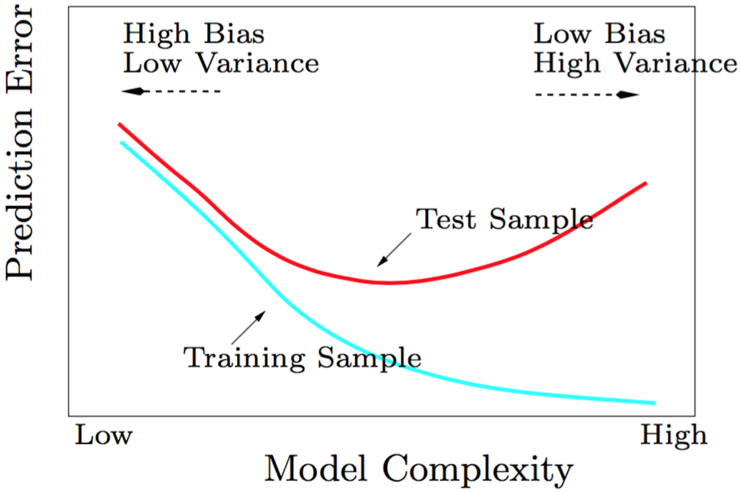


Figure 1: Bias Variance tradeoff for some model family. Example: model complexity for the k-nearest-neighbor family of methods is the parameter k.

With larger model complexity, the model adapts itself too much to the training data, leading to overfitting. On the other hand, if the model is not complex enough, it will lead to underfitting.

2 Linear Regression (Continued)

Recall that

$$\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\,\mathbf{y}$$

minimizes RSS(β).

2.1 Sampling Properties of $\widehat{\beta}$

Now we bring in the joint distribution and use it to describe properties of $\widehat{\beta}$. Lets assume:

- y_i are uncorrelated and have variance σ^2.
- x_i are non-random.

Under the above assumptions, $Var(\widehat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$

To make further inference, lets assume $E(Y|X) = X^T\beta$ and $Y = X^T\beta + \epsilon$, with $\epsilon \sim N(0, \sigma^2)$. Then,

$$\widehat{\beta} \sim N(\beta, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2)$$
$$\widehat{\sigma} \sim \frac{\sigma^2}{N-p}\chi^2_{N-p}$$

Note 1. In the ESLII book, the assume $\mathbf{X}$ is $p+1$ dimensional, so there is a slight change in the above formula.

This allows us to define some hypothesis tests and confidence intervals for β and β_j.

1. Let *Z-score* be $\frac{\widehat{\beta}}{\widehat{\sigma}\sqrt{v_j}}$, where v_j is the j^{th} diagonal entry of $(\mathbf{X}^T\mathbf{X})^{-1}$. Under the null that β_j is 0, z_j is distributed as t_{N-p-1} (t distribution). If we know σ, z_j will be a standard normal.

2. To test for groups of variables (e.g., if the original X_j is categorical, then several variables X_k will be related): We define the F statistic as:

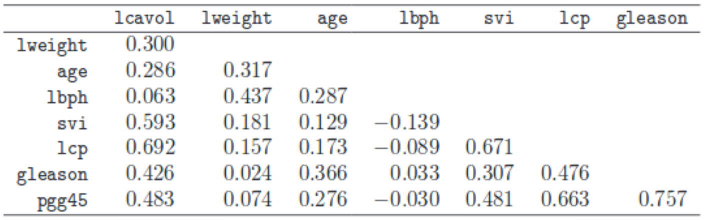
$$F = \frac{(RSS_0 - RSS_1)/(p_1 - p_0)}{RSS_1/(N-p_1)},$$

   where RSS_1 is for the bigger model. This statistic will be $F_{p_1-p_0, N-p_1}$ distributed under the null that the smaller model is correct.

3. Confidence intervals: Isolating β_j, we get a $1-2\alpha$ confidence interval as

$$\{\widehat{\beta} - z^{1-\alpha}\sqrt{v_j}\,\widehat{\sigma}, \widehat{\beta} + z^{1-\alpha}\sqrt{v_j}\,\widehat{\sigma}\}.$$

   For instance, $z^{1-0.05} = 1.645$ for standard normal.



	lcavol	lweight	age	lbph	svi	lcp	gleason
lweight	0.300						
age	0.286	0.317					
lbph	0.063	0.437	0.287				
svi	0.593	0.181	0.129	−0.139			
lcp	0.692	0.157	0.173	−0.089	0.671		
gleason	0.426	0.024	0.366	0.033	0.307	0.476	
pgg45	0.483	0.074	0.276	−0.030	0.481	0.663	0.757

Figure 2: Prostate cancer dataset: predictor correlations.

2.2 Prostate Cancer Example

Figure 2 and 3 shows how the input variables are correlated. These are: log cancer volume (lcavol), log prostate weight (lweight), age, log of benign prostate hyperplastia (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason) and percent of Gleason scores 4 or 5 (pgg45). The response variable is the level of prostate-specific antigen (lpsa).

We can fit a linear regression model to this data after standardizing the inputs. Training has 67 observations and test has 30. The output of regression is shown in Figure 4. With 9 parameters, the 0.025 quantile of t_{67-9} is ± 2. Notice that lcp and lcavol are strongly correlated but only the latter is significant.

The prediction error is 0.521. Predicting using the mean value of lpsa would have gotten 1.057 (the *base error rate*). The linear model reduces this error by about 50%.

Most models are distortion of truth. A good model is obtained when a suitable trade-off is made between bias and variance.

Note 2. Mean Squared Error (MSE) is related to Expected (Squared) Prediction Error (EPE) at a point x_0. For instance, for estimate $x_0^T\widehat{\beta}$, $E(Y_0 - x_0^T\widehat{\beta}) = \sigma^2 + E(x_0^T\widehat{\beta} - x_0^T\beta)^2$.

Example 2. If X is 1-dimensional and there is no intercept, then $\widehat{\beta} = \frac{\sum_{i=1}^N x_i y_i}{\sum_{i=1}^N x_i^2}$.(Check that this is true yourself!)

Example 3. Say the columns of input data matrix $\mathbf{X}$, $\mathbf{x}_j$ are orthogonal. Then, each coefficient $\widehat{\beta}_j = \frac{\mathbf{x}_j^T\mathbf{y}}{\mathbf{x}_j^T\mathbf{x}}$. You can get this by expanding out the closed form formula and using the fact that $\mathbf{x}_i^T\mathbf{x}_j = 0$ for $i \neq j$.

Note 3. What happens if Y is K-dimensional? Well, clearly β is not a vector anymore. It is a $p \times K$ dimensional matrix and the model is $Y = X\beta + E$ (E is the error matrix). It turns out that the least squares estimated matrix $\widehat{\beta}$ is still the same as before: i.e., $\widehat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. What this means is that the coefficients for the k^{th} outcome/output variable are just the least squares estimates when the k^{th} coordinate is regressed on X. In other words, it is as if you are solving K different single-dimensional-output linear regression problems.

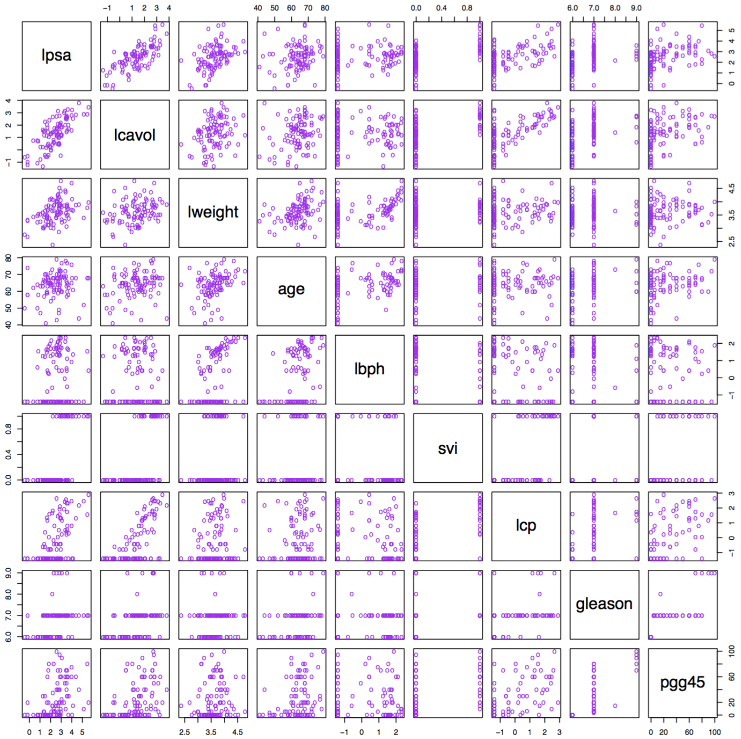


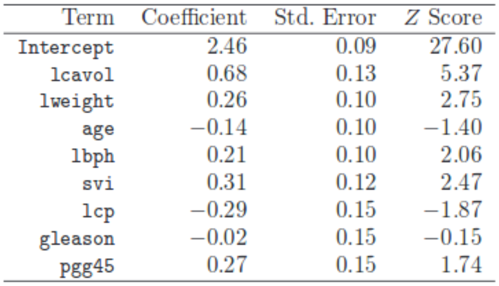
Figure 3: Scatterplot of predictors.

Least squares is great, especially due to its rich history and use in practice. But what happens when: (a) the prediction accuracy is not good enough? And (b) there are lots of feature coordinates? We will see that introducing more bias can sometimes improve prediction accuracy as well as tell us about feature importance. This is what we will do next!

Example 4. When there are many correlated input variables, it is empirically observed that linear regression coefficients become poorly determined and exhibit high variance. A large positive coefficient on one coordinate can be cancelled by a similarly large negative coefficient on another related coordinate.

3 Biasing Linear Regression via Subset Selection, Ridge and LASSO

We will introduce bias to linear regression via three methods: subset selection, ridge regression and LASSO. The bias will be such that some input variables will have higher coefficients, allowing us to infer that these are the important ones. This is related to *model selection*, which is a topic that will be discussed later.



Term	Coefficient	Std. Error	Z Score
`Intercept`	2.46	0.09	27.60
`lcavol`	0.68	0.13	5.37
`lweight`	0.26	0.10	2.75
`age`	−0.14	0.10	−1.40
`lbph`	0.21	0.10	2.06
`svi`	0.31	0.12	2.47
`lcp`	−0.29	0.15	−1.87
`gleason`	−0.02	0.15	−0.15
`pgg45`	0.27	0.15	1.74

Figure 4: Prostate cancer dataset: Linear regression output.

3.1 Subset Selection

What do we want to do here? We want to retain only a subset of input variables and discard the rest of the input variables.

How do we do this?

- Strategy 1: Say the number of features is p. Then for every $1 \leq k \leq p$, we find the RSS with linear regression with just k features. Note that for $\binom{p}{k}$ such subsets of input variables to consider. See Figure 5 for a plot of RSS for the prostate cancer example.

The best-subset curve (red line in Figure 5) is decreasing, so it cannot be used to pick k. The choice of k depends on bias and variance. One way to choose is: we pick k that minimizes an estimate of error. Such an estimate can be obtained by *cross validation* (see below).

- Strategy 2: Because considering all subsets is time consuming[1], one could start with the intercept and sequentially add a variable that reduces the RSS the most. This is called a *greedy* procedure, and has a couple of advantages: We only consider $O(p^2)$ subsets instead of $O(2^p)$, implying less computation. Further, lesser subsets to consider also implies more bias and less variance[2].

Example 5. Although we omit the details here, both strategy 1 and strategy 2 above give the same subset.

When we do subset selection and get the input variables that seem to be the best in predicting the dependent variable, we should avoid displaying the standard errors and z-scores for the corresponding model. Why is this the case? This is because data was not just used to get the model $\widehat{\beta}$, but also used to subset features. This makes the regression analysis we saw before, inapplicable!

[1] The number of all non-empty subsets of a set with p elements is $2^p - 1$.

[2] This is a hand-wavy remark for now, we haven't made this precise.

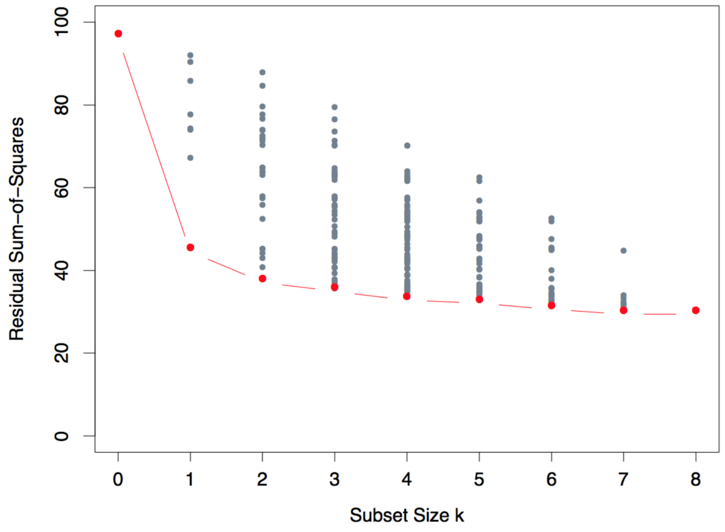


Figure 5: Prostate cancer dataset: Subset selection.

3.1.1 Cross Validation Primer

Term	LS	Best Subset	Ridge	Lasso	PCR	PLS
`Intercept`	2.465	2.477	2.452	2.468	2.497	2.452
`lcavol`	0.680	0.740	0.420	0.533	0.543	0.419
`lweight`	0.263	0.316	0.238	0.169	0.289	0.344
`age`	−0.141		−0.046		−0.152	−0.026
`lbph`	0.210		0.162	0.002	0.214	0.220
`svi`	0.305		0.227	0.094	0.315	0.243
`lcp`	−0.288		0.000		−0.051	0.079
`gleason`	−0.021		0.040		0.232	0.011
`pgg45`	0.267		0.133		−0.056	0.084
Test Error	0.521	0.492	0.492	0.479	0.449	0.528
Std Error	0.179	0.143	0.165	0.164	0.105	0.152

Figure 6: Prostate cancer dataset: Coefficients estimated using extensions of linear regression. The last two lines are post cross-validation and report numbers using the 30 observation test data.

Figure 6 shows the performance of subset selection compared to vanilla linear regression (there are a few other methods there, and we will discuss them soon).

Lets go into the details of cross-validation that we mentioned earlier. It is a useful tool to get an estimate of test error, which itself is useful to pick the right subset size.

Note 4. Why do we need an estimate of test error? We need it to find the right subset size k, for instance. And we should not use the original test data for this. Because that data is to score the final model. The choice of k should only be determined using the data you are working with, i.e., training data!

Cross validation happens as follows. You want to estimate test error for a k that you chose. You do the following:

- Break training data randomly into 10 buckets[3] (sets or *folds*).
- For a given k, do the following for each subset:
    - Compute the linear model using the subset of input features, using 9 of the folds. Evaluate the performance of the corresponding model on the 10th fold.
    - Repeat the above step in-turn for each of the 10 folds.
    - Average the errors. This is your estimate of the prediction error for this subset.
- Find the best subset of size k by doing the above for all subsets of the given size k, and picking the subset with the lowest estimated prediction error.
- Do the above for each k to get the corresponding error estimates, one per value of k. Now pick the best k (and the subset) using these error estimates.

Once the k and the subset is picked, you could use the full training data to build the final regression model. This can then be scored against the original test data (e.g., in the prostate cancer example, we had 67 observations in training and 30 observations as part of the test data).

4 Summary

We learned the following things:

- Bias variance trade-off.
- Linear Regression: sampling properties of $\widehat{\beta}$.
- Subset selection.

In the next lecture, we will discuss ridge regression, LASSO, and the nuances of classification through: (a) Linear Discriminant Analysis (LDA), and (b) logistic regression.

[3] The choice of 10 was arbitrary here.

A Sample Exam Questions

1. Is k-nearest neighbor the best method among all supervised learning methods? Explain why or why not.

2. How would you use a categorical feature (say, taking three values (cat,dog,fox)) in linear regression?

3. In which situations would $\mathbf{X}$ not be full rank? Does this pose any issue in linear regression modeling? If so, how would you mitigate it?

4. How would you perform a hypothesis test to check the null $\beta_j = 0$?