Fundamentals of Data Science

Model scores

Ramesh Johari

Model scores

In this lecture we develop an approach to estimation of prediction using limited data (i.e., "in-sample" estimation of prediction error), that relies on underlying assumptions about the model that generated the data.

Model scores

Model scoring uses the following approach:

- Choose a model, and fit it using the data.
- Compute a *model score* that uses the sample itself to estimate the prediction error of the model.

By necessity, this approach works only for certain model classes; we show how model scores are developed for linear regression.

Training error

The first idea for estimating prediction error of a fitted model might be to look at the sum of squared error in-sample:

$$\mathsf{Err}_{\mathsf{tr}} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}(\mathbf{X}_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\hat{r}_i^2.$$

This is called the *training error*; it is the same as $1/n$ $\times$ *sum of squared residuals* we studied earlier.

Training error vs. prediction error

Of course, we should expect that training error is *too optimistic* relative to the error on a new test set: after all, the model was specifically tuned to do well on the training data.

To formalize this, we can compare $\mathsf{Err}_{\mathsf{tr}}$ to $\mathsf{Err}_{\mathsf{in}}$, the *in-sample prediction error*:

$$\mathsf{Err}_{\mathsf{in}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[(Y - \hat{f}(\vec{X}))^2 | \mathbf{X}, \mathbf{Y}, \vec{X} = \mathbf{X}_i].$$

This is the prediction error if we received new samples of Y corresponding to each covariate vector in our existing data.[1]

[1] The name is confusing: "in-sample" means that it is prediction error on the covariate vectors $\mathbf{X}$ already in the training data; but note that this measure is the expected prediction error on *new* outcomes for each of these covariate vectors.

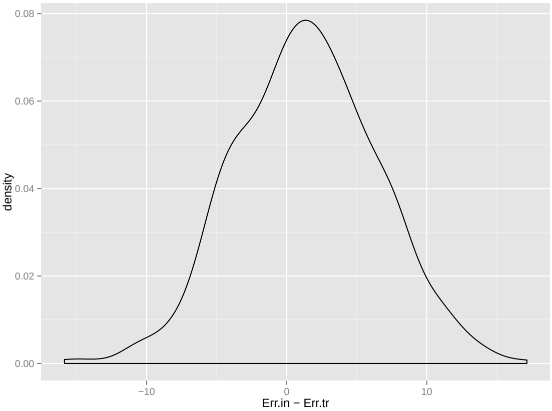
Training error vs. test error

Let's first check how these behave relative to each other.

- Generate 100 $X_1, X_2 \sim N(0,1)$, i.i.d.
- Let $Y_i = 1 + X_{i1} + 2X_{i2} + \varepsilon_i$, where $\varepsilon_i \sim N(0,5)$, i.i.d.
- Fit a model $\hat{f}$ using OLS, and the formula `Y ~ 1 + X1 + X2`.
- Compute training error of the model.
- Generate another 100 *test samples* of Y corresponding to each row of $\mathbf{X}$, using the same population model.
- Compute in-sample prediction error of the fitted model on the test set.
- Repeat this process 500 times, and create a plot of the results.

Training error vs. test error

Results:



Mean of $\text{Err}_{\text{in}} - \text{Err}_{\text{tr}} = 1.42$; i.e., training error is underestimating in-sample prediction error.

Training error vs. test error

If we could somehow *correct* Err_{tr} to behave more like Err_{in}, we would have a way to estimate prediction error on new data (at least, for covariates $\mathbf{X}_i$ we have already seen).

Here is a key result towards that correction.[2]

Theorem

$$\mathbb{E}[\text{Err}_{\text{in}}|\mathbf{X}] = \mathbb{E}[\text{Err}_{\text{tr}}|\mathbf{X}] + \frac{2}{n}\sum_{i=1}^{n}\text{Cov}(\hat{f}(\mathbf{X}_i), Y_i|\mathbf{X}).$$

In particular, if $\text{Cov}(\hat{f}(\mathbf{X}_i), Y_i|\mathbf{X}) > 0$*, then training error underestimates test error.*

[2] This result holds more generally for other measures of prediction error, e.g., 0-1 loss in binary classification.

Training error vs. test error: Proof [$*$]

Proof: If we expand the definitions of $\mathsf{Err}_{\mathsf{tr}}$ and $\mathsf{Err}_{\mathsf{in}}$, we get:

$$\mathsf{Err}_{\mathsf{in}} - \mathsf{Err}_{\mathsf{tr}} = \frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}[Y^2|\vec{X} = \mathbf{X}_i] - Y_i^2 \\ - 2(\mathbb{E}[Y|\vec{X} = \mathbf{X}_i] - Y_i)\hat{f}(\mathbf{X}_i)\Big)$$

Now take expectations over $\mathbf{Y}$. Note that:

$$\mathbb{E}[Y^2|\mathbf{X}, \vec{X} = \mathbf{X}_i] = \mathbb{E}[Y_i^2|\mathbf{X}],$$

since both are the expectation of the square of a random outcome with associated covariate $\mathbf{X}_i$. So we have:

$$\mathbb{E}[\mathsf{Err}_{\mathsf{in}} - \mathsf{Err}_{\mathsf{tr}}|\mathbf{X}] = -\frac{2}{n}\sum_{i=1}^{n}\mathbb{E}\Big[(\mathbb{E}[Y|\vec{X} = \mathbf{X}_i] - Y_i)\hat{f}(\mathbf{X}_i)\Big|\mathbf{X}\Big].$$

Training error vs. test error: Proof [$*$]

Proof (continued): Also note that $\mathbb{E}[Y|\vec{X} = \mathbf{X}_i] = \mathbb{E}[Y_i|\mathbf{X}]$, for the same reason. Finally, since:

$$\mathbb{E}[Y_i - \mathbb{E}[Y_i|\mathbf{X}]|\mathbf{X}] = 0,$$

we get:

$$\mathbb{E}[\mathsf{Err}_{\mathsf{in}} - \mathsf{Err}_{\mathsf{tr}}|\mathbf{X}] = \frac{2}{n}\sum_{i=1}^{n}\Big(\mathbb{E}\Big[(Y_i - \mathbb{E}[Y|\vec{X} = \mathbf{X}_i])\hat{f}(\mathbf{X}_i)\Big|\mathbf{X}\Big] \\ - \mathbb{E}[Y_i - \mathbb{E}[Y_i|\mathbf{X}]|\mathbf{X}]\mathbb{E}[\hat{f}(\mathbf{X}_i)\big|\mathbf{X}]\Big),$$

which reduces to $(2/n)\sum_{i=1}^{n}\mathrm{Cov}(\hat{f}(\mathbf{X}_i), Y_i|\mathbf{X})$, as desired.

The theorem's condition

What does $\mathrm{Cov}(\hat{f}(\mathbf{X}_i), Y_i | \mathbf{X}) > 0$ mean?

In practice, for any "reasonable" modeling procedure, we should expect our predictions to be positively correlated with our outcome.

Example: Linear regression

Assume a linear population model $Y = \vec{X}\boldsymbol{\beta} + \varepsilon$, where $\mathbb{E}[\varepsilon|\vec{X}] = 0$, $\mathrm{Var}(\varepsilon) = \sigma^2$, and errors are uncorrelated.

Suppose we use a subset S of the covariates and fit a linear regression model by OLS. Then:

$$\sum_{i=1}^{n} \mathrm{Cov}(\hat{f}(\mathbf{X}_i), Y_i|\mathbf{X}) = |S|\sigma^2.$$

In other words, in this setting we have:

$$\mathbb{E}[\mathsf{Err}_{\mathsf{in}}|\mathbf{X}] = \mathbb{E}[\mathsf{Err}_{\mathsf{tr}}|\mathbf{X}] + \frac{2|S|}{n}\sigma^2.$$

A model score for linear regression

The last result suggests how we might estimate in-sample prediction error for linear regression:

- Estimate σ^2 using the sample standard deviation of the residuals on the full fitted model, i.e., with $S = \{1, \dots, p\}$; call this $\hat{\sigma}^2$.[3]
- For a given model using a set of covariates S, compute:

$$C_p = \mathsf{Err}_{\mathsf{tr}} + \frac{2|S|}{n}\hat{\sigma}^2.$$

This is called *Mallow's C_p statistic*. It is an estimate of the prediction error.

[3]Informally, the reason to use the full fitted model is that this should provide the best estimate of σ^2.

A model score for linear regression

$$C_p = \mathsf{Err}_{\mathsf{tr}} + \frac{2|S|}{n}\hat{\sigma}^2.$$

How to interpret this?

- The first term measures fit to the existing data.
- The second term is a penalty for *model complexity*.

So the C_p statistic balances underfitting and overfitting the data; for this reason it is sometimes called a *model complexity score*.

(We will later provide conceptual foundations for this tradeoff in terms of *bias* and *variance*.)

AIC, BIC

Other model scores:

- *Akaike information criterion* (AIC). In the linear population model with *normal* ε, this is equivalent to:

$$\frac{n}{\hat{\sigma}^2}\left(\mathsf{Err}_{\mathsf{tr}} + \frac{2|S|}{n}\hat{\sigma}^2\right).$$

- *Bayesian information criterion* (BIC). In the linear population model with normal ε, this is equivalent to:

$$\frac{n}{\hat{\sigma}^2}\left(\mathsf{Err}_{\mathsf{tr}} + \frac{|S|\ln n}{n}\hat{\sigma}^2\right).$$

Both are more general, and derived from a *likelihood* approach. (More on that later.)

AIC, BIC

Note that:

- AIC is the same (up to scaling) as C_p in the linear population model with normal ε.
- BIC penalizes model complexity more heavily than AIC.

AIC, BIC in software [$*$]

In practice, there can be significant differences between the actual values of C_p, AIC, and BIC depending on software; but these don't affect model selection.

- The estimate of sample variance $\hat{\sigma}^2$ for C_p will usually be computed using the full fitted model (i.e., with all p covariates), while the estimate of sample variance for AIC and BIC will usually be computed using just the fitted model being evaluated (i.e., with just $|S|$ covariates). This typically has no substantive effect on model selection.
- In addition, sometimes AIC and BIC are reported as the *negation* of the expressions on the previous slide, so that larger values are better; or without the scaling coefficient in front. Again, none of these changes affect model selection.

Comparisons

Simulation: Comparing C_p, AIC, BIC, CV

Repeat the following steps 10 times:

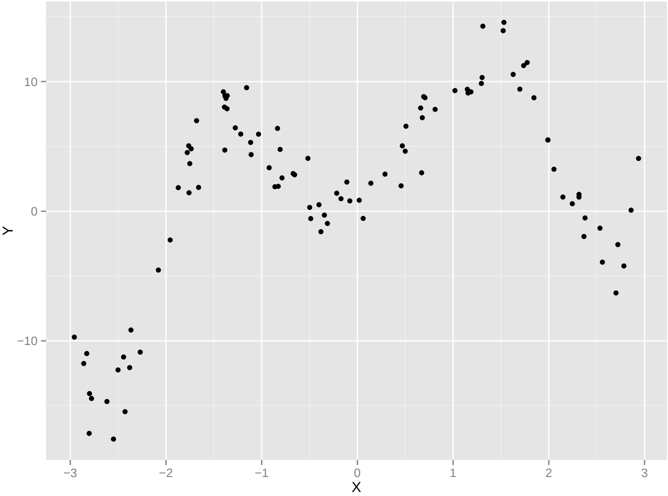
- For $1 \leq i \leq 100$, generate $X_i \sim$ uniform$[-3, 3]$.
- For $1 \leq i \leq 100$, generate Y_i as:

$$Y_i = \alpha_1 X_i + \alpha_2 X_i^2 - \alpha_3 X_i^3 + \alpha_4 X_i^4 - \alpha_5 X_i^5 + \alpha_6 X_i^6 + \varepsilon_i,$$

  where $\varepsilon_i \sim$ uniform$[-3, 3]$.
- For $p = 1, \ldots, 20$, we evaluate the model `Y ~ 0 + X + I(X^2) + ... + I(X^p)` using C_p, AIC, BIC, and 10-fold cross validation.

How do these methods compare?

Simulation: Visualizing the data



Simulation: Comparing C_p, AIC, BIC, CV