BIOS 6312: Modern Biostatistics Methodology II

Andrew J. Spieker, Ph.D.

Associate Professor of Biostatistics
Vanderbilt University

Set 10: Predictive capacity of regression models

Version: 04/26/2025

Table of Contents

Framing prediction problems

Ideas:

- Much of our focus has been on interpretation and estimations of coefficients. In this unit, our goals are are about prediction:
  - $\mathrm{E}[Y|\mathbf{X} = \mathbf{x}] = \mathbf{x}^T\boldsymbol{\beta}$ (linear model).
  - Predicted mean: $\widehat{Y}(\mathbf{x}) = \mathbf{x}^T\widehat{\boldsymbol{\beta}}$.
  - We even discussed prediction *intervals* for continuous outcomes. Our current discussion of prediction is a bit different in the sense that we want to learn about how close $\widehat{Y}(\mathbf{x})$ is to $\mathrm{E}[Y|\mathbf{X} = \mathbf{x}]$.
- Key questions:
  - How do we know how well a method has done?
  - How do we compare methods?
- Questions for another course (e.g., statistical learning):
  - How do we improve our models without overfitting the data?

Framing prediction problems

Defining prediction error: Metrics

- Example: $\mathrm{E}[Y|\mathbf{X} = \mathbf{x}] = \mathbf{x}^T\boldsymbol{\beta}$ (linear model).
  - Predicted mean: $\widehat{Y}(\mathbf{x}) = \mathbf{x}^T\widehat{\boldsymbol{\beta}}$.
- Many ways we could think to evaluate predictive ability.
  - Mean squared error: $\mathrm{E}[(Y - \widehat{Y})^2]$.
  - Mean absolute error: $\mathrm{E}[|Y - \widehat{Y}|]$.
- In the two above examples, lower values indicate better predictive ability (difficult to compare methods across different prediction error metrics).
- For now, let's focus on mean squared error, $\mathrm{E}[(Y - \widehat{Y})^2]$.

Framing prediction problems

Improving prediction error: Naive method

- Possible method: include more covariates.
- Example: MRI data
  - Y: DSST
  - X: one covariate? A whole bunch of covariates?
- Compare root mean squared error (RMSE) between models of differential complexity.

Improving prediction error: Naive method

```
## Read in data set
dat <- read.csv("mri.csv")

## Fit model (output not relevant for the current goal)
model <- regress("mean", dsst ~ male, data = dat)

## RMSE (relevant for the current goal)
> model$sigma
[1] 12.64
```

Improving prediction error: Naive method

```
## Characterize factor variables as such
dat$race <- factor(dat$race)
dat$chf <- factor(dat$chf)
dat$genhlth <- factor(dat$genhlth)
dat$numinf <- factor(dat$numinf)

## Fit model (output not relevant for the current goal)
model <- regress("mean", dsst ~ male + age + race + weight +
                 height + packyrs + yrsquit + alcoh +
                 physact + chf + diabetes + genhlth +
                 ldl + alb + crt + sbp + aai + fev +
                 atrophy + numinf, data = dat)

## RMSE (relevant for the current goal)
> model$sigma
[1] 10.98
```

Improving prediction error: Naive method

- This method suffers from the fact that a reduced RMSE could be explained by the fact that your model...
  - ...better predicts the outcome in the population.
  - ...merely serves as a better fit to *your data*.
  - ...or a combination of these two!
- Not interesting to know well your model predicts the observations used to fit that model. Of greater interest: knowing how well model predicts observations *not* in your sample.

Table of Contents

The bias-variance trade-off

Prediction error: Bias-variance decomposition

- Consider general case: $\mathrm{E}[Y|\mathbf{X} = \mathbf{x}] = f(\mathbf{x})$.
- Estimate f on a *training* set to obtain $\widehat{f}$.
- Consider random out-of-sample observation, $(\mathbf{X}_0, Y_0)$.
- Mean squared error at $\mathbf{X}_0$ is:

$$\begin{aligned}\mathrm{E}[(Y_0 - \widehat{f}(\mathbf{X}_0))^2] &= \mathrm{E}[(Y_0 - \mathrm{E}[Y_0])^2] + \mathrm{E}[(\mathrm{E}[Y_0] - \mathrm{E}[\widehat{f}(\mathbf{X}_0)])^2] \\ &\quad + \mathrm{E}[(\widehat{f}(\mathbf{X}_0) - \mathrm{E}[\widehat{f}(\mathbf{X}_0)])^2] \\ &= \sigma^2_{Y_0} + \mathrm{Bias}^2(\widehat{f}(\mathbf{X}_0)) + \mathrm{Var}(\widehat{f}(\mathbf{X}_0)).\end{aligned}$$

- Prediction error: trade-off between bias and variance.
  - Bias: Model *correctly* captures $\mathbf{x}/\mathrm{E}[Y|\mathbf{X} = \mathbf{x}]$ relationship?
  - Variance: Model *precisely* captures relationship?
- *Training* error (internal) does not estimate the *test* error (external) well because it does not properly account for model complexity.

The bias-variance trade-off

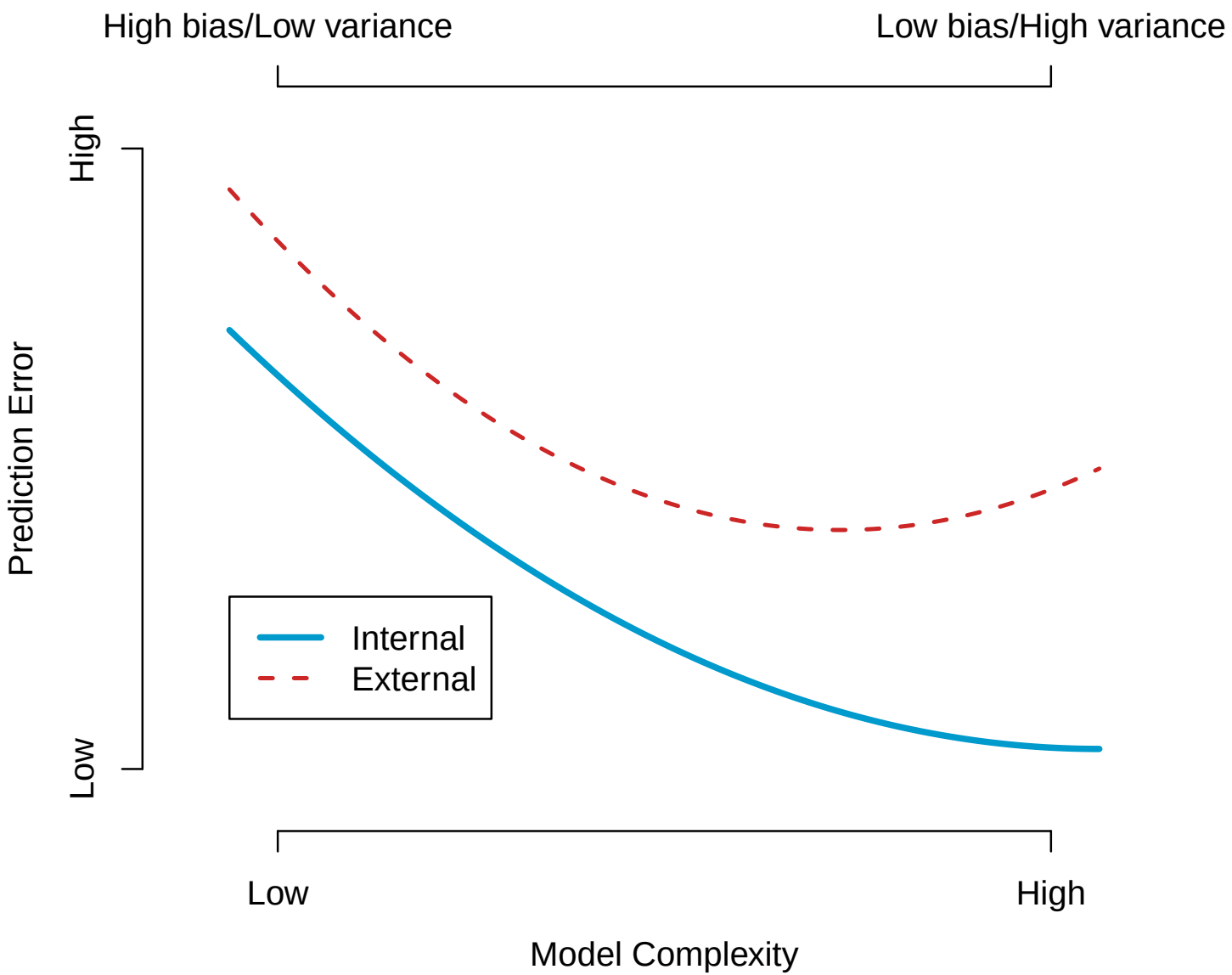

Table of Contents

Training and test error

Estimating test error: Simple method

1. Split your sample (say, 80/20); label the larger portion as *training* set and the other as a *test* set.
2. Fit your model on the training set.
3. Use fitted model to predict the values on the test set and estimate the test prediction error.

Idea: Mimic the process of getting two independent sets of data.

R: Sample splitting

```r
## Sample size
n <- dim(dat)[1]

## Set see for reproducibility
set.seed(6312)

## Random split
samp <- sample(c(rep(0, 588), rep(1,147)), replace = FALSE)
dat0 <- dat[samp == 0,]
dat1 <- dat[samp == 1,]
```

R: Training and test error for simpler model

```
## Fit simpler model
model <- regress("mean", dsst ~ male, data = dat0)

## Training error
> model$sigma
[1] 12.58

## Test error
MSE.test <- mean((dat1$dsst - predict(model, newdata = dat1)
    [,1])^2, na.rm = TRUE)
sigma.test <- sqrt(MSE.test)

> sigma.test
[1] 12.91
```

Training and test error

R: Training and test error for more complex model

```
## Fit more complex model
model <- regress("mean", dsst ~ male + age + race + weight +
                   height + packyrs + yrsquit + alcoh +
                   physact + chf + diabetes + genhlth +
                   ldl + alb + crt + sbp + aai + fev +
                   atrophy + numinf, data = dat0)

## Training error
> model$sigma
[1] 10.93

## Test error
MSE.test <- mean((dat1$dsst - predict(model, newdata = dat1)
    [,1])^2, na.rm = TRUE)
sigma.test <- sqrt(MSE.test)

> sigma.test
[1] 11.44
```

Training and test error

Example: Training/test error for DSST (MRI data)

- Results (RMSE).

	Training	**Test**
Single predictor	12.6	12.9
Many predictors	10.9	11.4

- What do you notice?

Training and test error

Ideas: Model complexity in linear models

- Model complexity can be characterized by the number of degrees of freedom used by the model, df_M.
- Linear regression:

$$df_M = \frac{1}{\sigma^2} \sum_{i=1}^{N} \text{Cov}(\widehat{Y}_i, Y_i).$$

- With K predictors plus an intercept, $df_M = K + 1$.
  - Absent an intercept, there are K degrees of freedom.
- Highly complex models tend to have lower bias as compared to less complex models, and will provide better prediction error on the *training* data used to fit the model.

Table of Contents

Prediction for binary outcomes

Binary outcomes:

- Logistic regression can be used to characterize how well a set of predictors can correctly classify an outcome.
- Model:

$$\text{logit}(\text{P}(Y = 1|\mathbf{X} = \mathbf{x})) \quad = \quad \mathbf{x}^T\boldsymbol{\beta}.$$

- Predicted values:

$$\widehat{y}_i = \widehat{Y}(\mathbf{x}_i) = \widehat{\text{P}}(Y = 1|\mathbf{X} = \mathbf{x}_i) = \text{expit}(\mathbf{x}_i^T\boldsymbol{\beta}).$$

- General idea: Those with a true value of $Y = 1$ should have higher predicted values than those with a true value of $Y = 0$.

Prediction for binary outcomes

Binary outcomes:

- We can use the receiver operating characteristic (ROC) curve to characterize the classification ability.
- Consider a cutoff point, c (with $0 \leq c \leq 1$), above which all with $\widehat{Y} > c$ are classified as $Y_c = 1$ on the basis of the logistic regression model.
- Can characterize two metrics of predictive ability:
  - $P(Y_c = 0|Y = 0) = P(\widehat{Y} \leq c|Y = 0)$: proportion classified as $Y = 0$ among those with a true value of $Y = 0$.
  - $P(Y_c = 1|Y = 1) = P(\widehat{Y} > c|Y = 1)$: proportion classified as $Y = 1$ among those with a true value of $Y = 1$.
- These correspond to the specificity and sensitivity.
- ROC curve: plot of (1− specificity) and sensitivity across different values of c.

Prediction for binary outcomes

Binary outcomes:

- Area under the ROC curve (AUC) summarizes the predictive ability in a single quantity.
- An AUC of 1.00 corresponds to perfect classification; an AUC of 0.50 signifies no classification ability.
- Note: AUC corresponds to a scaled variant of Wilcoxon Rank Sum test statistic, and estimates $\mathrm{P}(\widehat{y}^1 > \widehat{y}^0)$, where, e.g., $\widehat{y}^1$ denotes the predicted probability (of $Y = 1$) for a randomly sampled individual with $Y = 1$.

Prediction for binary outcomes

Example: Diabetes in MRI data

- Y: Diabetes
- Predictors: age, sex, smoking history (pack years), physical activity, weight, height.
- Model:

$$\text{logit}(\text{P}(Y = 1|\mathbf{X} = \mathbf{x})) = \mathbf{x}^T\boldsymbol{\beta}.$$

- Naturally, we want to divide the data into a training and test set in order to characterize predictive ability.

R: Training and test AUC

```
## Fit model
model <- regress("odds", diabetes ~ age + male + packyrs +
                   physact + weight + height, data = dat0)

## TRAINING ERROR
## Predicted values
y0hat <- predict(model, type = "response")

## Clever way to get the AUC
y0hat.1 <- y0hat[dat0$diabetes == 1]
y0hat.0 <- y0hat[dat0$diabetes == 0]
n1 <- length(y0hat.1)
n0 <- length(y0hat.0)
AUC0 <- (wilcox.test(y0hat.1,y0hat.0)$statistic)/(n0 * n1)
> as.numeric(AUC0)
[1] 0.6881

## TEST ERROR (Totally analogous)
y1hat <- predict(model, newdata = dat1)
y1hat.1 <- y1hat[dat1$diabetes == 1]
y1hat.0 <- y1hat[dat1$diabetes == 0]
n1 <- length(y1hat.1)
n0 <- length(y1hat.0)
AUC1 <- (wilcox.test(y1hat.1,y1hat.0)$statistic)/(n0 * n1)
> as.numeric(AUC1)
[1] 0.6136
```

Prediction for binary outcomes

Example: Diabetes in MRI data

- Unsurprisingly, training AUC higher relative to the test AUC.
- There are many other metrics of predictive ability that can be applied to binary outcomes.

Summary

This unit:

- The bias-variance trade-off.
- Training error does not estimate test error.
- Measures of predictive ability (continuous/binary)
- A lot let unsaid.
  - Forward- and backward-selection.
  - Model optimism.
  - Penalized regression.
  - Regression trees.
  - Support vector machines.
  - Neural nets.
- This unit is just to orient you to the basics.

Summary

So far:

- Review.
- Simple linear regression.
- Multiple linear regression (foundations).
- Multiple linear regression (interactions and strata).
- Transformations and basis expansions.
- Regression with binary outcomes.
- Regression with nominal, ordinal, and count outcomes.
- Introduction to clustered data.
- Methods for time-to-event outcomes.
- Predictive capacity of regression models.

Summary

The end!

- Comprehensive examinations for biostatistics students.
- Summer research.
- For those taking advanced regression: Review your linear algebra!
  - Eigenvalues and eigenvectors.
  - Projection matrices.
  - Positive definite matrices.
  - Singular value decomposition.
- Don't stop learning!