CWRU DSCI353-353M-453: 02b Intro to LinRegr-ISLR2

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

26 January, 2023

Contents

2.2.2.1 Class Readings, Assignments, Syllabus Topics

2.2.2.1.1 Reading, Lab Exercises, SemProjects

- Readings:
  - For today: DL01, DL02, (R4DS7-8)
  - For next class: DL03, ISLR3
- Laboratory Exercises:
  - LE1 Given out today
  - LE1 is due on Thursday Feb. 2nd
- Office Hours: (Class Canvas Calendar for Zoom Link)

- Wednesdays @ 4:00 PM to 5:00 PM
- Saturdays @ 3:00 PM to 4:00 PM
- **Office Hours are on Zoom, and recorded**

- Semester Projects
  - DSCI 453 Students Biweekly Updates Due
    - Update #1 is Due ** This Friday **
  - DSCI 453 Students
    - Next Report Out #1 is Due ** Feb. '17th **
  - All DSCI 353/353M/453, E1453/2453 Students:
    - Peer Grading of Report Out #1 is Due ** **
  - Exams
    - MidTerm: **Thursday March 9th**, in class or remote, 11:30 - 12:45 PM
    - Final: **Thursday May 4th**, 2023, 12:00PM - 3:00PM, Nord 356 or remote

2.2.2.1.2 Textbooks

- Introduction to R and Data Science
  - For R, Coding, Inferential Statistics
    - Peng: R Programming for Data Science
    - Peng: Exploratory Data Analysis with R

Textbooks for this class

- OIS = Diez, Barr, Çetinkaya-Runde: Open Intro Stat v4
- R4DS = Wickham, Grolemund: R for Data Science

Textbooks for DSCI353/353M/453, And in your Repo now

- ISLR = James, Witten, Hastie, Tibshirani: Intro to Statistical Learning with R
- ESL = Trevor Hastie, Tibshirani, Friedman: Elements of Statistical Learning
- DLwR = Chollet, Allaire: Deep Learning with R

Magazine Articles about Deep Learning

- DL1 to DL6 are "Deep Learning" articles in 3-readings/2-articles/

2.2.2.2 Syllabus

2.2.2.2.1 Tidyverse Cheatsheets, Functions and Reading Your Code

- Look at the Tidyverse Cheatsheet
  - **Tidyverse For Beginners Cheatsheet**
    - In the Git/20s-dsci353-353m-453-prof/3-readings/3-CheatSheets/ folder
  - **Data Wrangling with dplyr and tidyr Cheatsheet**

  Tidyverse Functions & Conventions

  - The pipe operator `%>%`
  - Use `dplyr::filter()` to subset data row-wise.
  - Use `dplyr::arrange()` to sort the observations in a data frame
  - Use `dplyr::mutate()` to update or create new columns of a data frame
  - Use `dplyr::summarize()` to turn many observations into a single data point
  - Use `dplyr::arrange()` to change the ordering of the rows of a data frame
  - Use `dplyr::select()` to choose variables from a tibble,
    - keeps only variables you mention
  - Use `dplyr::rename()` keeps all the variables and renames variables

Day:Date	Foundation	Practicum	Readings(optional)	Due(optional)
w01a:Tu:1/17/23	Markov Cluster	R, Rstudio IDE, Git		(LE0)
w01b:Th:1/19/23	Stat. Learning, Approach	Bash, Git, Class Repo	ISLR1,2 (R4DS-1-3)	
w02a:Tu:1/24/23	Lin. Regr. Bias-Var.	SemProjs; Regr. Ovrvw	ISLR3,(R4DS-4-6)	**(LE0:Due)** LE1
w02b:Th:1/26/23	Train/Test, Bias vs. Vari.	Tidyverse Review	DL01 DL02 (R4DS-7,8)	
w02Pr:Fr:1/27/23	**ADD DROP**	**DEADLINE**		**453 Update 1**
w03a:Tu:1/31/23	Logistic Regr. Classif	Tidy Wrangling	DL03,ISLR4	
w03b:Th:2/2/23	LDA	Multi-level Mod.	DL04, DL05	**LE1:Due**, LE2
w04a:Tu:2/7/23	Resample Cross-Valid.	Multilevel Mod.	ISLR5	
w04b:Th:2/9/23	Bootstrap	Mixed Effects		
w04Pr:Fr:2/10/23				**453 Update 2**
w05a:Tu:2/14/23	Subset Selec., Shrink.	Bootstrap	ISLR6 (R4DS9-16)	**LE2:Due**, LE3
w05b:Th:2/16/23	Mod. Selec. Dim. Red.	Clustering, ggplot2	DL06	
w05Pr:Fr:2/17/23				**453 Rep. Out 1**
w06a:Tu:2/21/23	Beyond Linear Modls	Feature Select., Caret	ISLR7, DL07	
w06b:Th:2/23/23	PCA, PCR, FA	Tidy Modeling	ISLR10(R4DS22-25)	**LE3:Due**, LE4
w06Pr:Fr:2/24/23				**453 Update 3**
w07a:Tu:2/28/23	Dec. Trees, Rand. Forest.	Machine Learning	ISLR8, DL08,09	
w07b:Th:3/2/23	MidTerm Review, SVM	SVM, SVR, ROC	ISLR9 (R4DS26-30)	**Peer Review 1**
w08a:Tu:3/7/23	R-Keras/TensorFlow2	Perceptron, Neural Nets	ISLR10	
w08b:Th:3/9/23	**MIDTERM EXAM**		DL10,11	**LE4:Due** LE5
w08Pr:Fr:3/10/23				**453 Update 4**
Tu:3/14/23	**SPRING**	**BREAK**	ISLR10	
Th:3/16/23	**SPRING**	**BREAK**	DL12,13	
w09a:Tu:3/21/23	Deep Learning	TF2 Keras Intro	Pocket Perceptron	ISLR10, DLR3
w09b:Th:3/23/23	Computer Vision, CNN	CNN w/TF2, Overfit	DLR4	
w09Pr:Fr:3/24/23				**453 Rep. Out 2**
w10a:Tu:3/28/23	Deep Learn Intro	NN Types	DLR5	
w10b:Th:3/30/23	DL CNN,RNN ImageNet	NN Types, CNN wTF2	Hinton ImageNet	
w10Pr:Fr:3/31/23				**453 Upd.5 & PrRev 2**
Sa:4/1/23				**LE5:Due** LE6
w11a:Tu:4/4/23	Fitting NNs	AUC,Prec,Recall Fruit		
w11b:Th:4/6/23	NLP, Graphs & ML		LeCun DL Rev. 2015	
w12a:Tu:4/11/23	Graphs & ML	NLP with sequences	DLR6	
w12b:Th:4/13/23	NLP w attention	Graph Repr Proc Wrkflw		**LE6:Due** LE7
w13a:Tu:4/18/23	DL Frameworks	Explaining DL w Lime		
w13b:Th:4/20/23	Linux Distros XGBoost	Explain Preds	Deep Dream	
w13Pr:Fr:4/21/23				**453 Rep. Out 3 Due**
w14a:Tu:4/25/23	Tranformers			
w14b:Th:4/27/23	Final Exam Review	Torch NN & DeepLearn		**LE7:Due**
w14Pr:Fr:4/28/23				**Peer Rev 3 Due**
	FINAL EXAM	**Th. 5/4/23, 12-3pm**	Nord 356 & Zoom	
	453 Final PDF Report	**Fr. 4/29, 11:59pm**		

Table 1: DSCI353-353M-453 Weekly Syllabus. R4DS-x.y, OISx.y, ISLRx.y, DLGBx.y refers to chapters and sections assigned as reading in our textbooks. DLx are deep learning articles.

Figure 1: Modeling, Prediction and Machine Learning Syllabus

    - rename(iris, petal_length = Petal.Length)
  - These can be combined using `dplyr::group_by()`
    - which lets you perform operations "by group".
  - The `%in%` matches conditions provided by a vector using the c() function
  - The **forcats** package has tidyverse functions
    - for factors (categorical variables)
  - The **readr** package has tidyverse functions
    - to read_…, melt_… col_…, parse_… data and objects

Reading Your Code: Whenever you see

- The assignment operator `<-`, think **"gets"**
- The pipe operator, `%>%`, think **"then"**

2.2.2.3 ISLR Chapter 2 Regression and IntroR Lab Excerise

- From Hastie and Tibshirani
  - They have good notation
  - And a good intro to R

2.2.2.3.1 Regression is the case of supervised learning

- Where we have a quantitative response
  - that is associated with the predictors
  - And we want to develop a predictive model
    - that relates predictors with response

2.2.2.4 Function Notation for a Predictive model

- Some notation for predictive models
  - Response Y which we want to predict
  - And the Predictors we will use are $\mathbf{X} = X_1 + X_2 + X_3$
    - when we have P number of predictors,
      - and $P = 3$ in this example
    - where the predictors $\mathbf{X}$ is a vector
    - And $\mathbf{X}$ is a column vector containing (X_1, X_2, X_3)
    - Which has 3 components $X_1 + X_2 + X_3$
    - We also have to have an error term ϵ
  - Our predictive model will then be
    - $Y = f(\mathbf{X}) + \epsilon$
  - ϵ error term is a catch all
    - captures measurement error, and other discrepancies
    - we can never model something perfectly
  - And for the predictor $\mathbf{X}$
    - A single instance of X is x
    - i.e. (x_1, x_2, x_3)
    - three specific values of the 3 components
    - of 1 individual observation, i.e. x
    - of the predictor X

2.2.2.4.1 Variables

- Independent Variables $\mathbf{X}$ are called
  - independent variables
  - predictors

    - exogenous variables
    - features (this is general CS term)
- Dependent Variables **Y** are called
    - dependent variables
    - responses
    - endogenous variables

In some cases, such as network models

- Some variables may be both.
    - independent, predictors
    - and also dependent response
- Such as in our group's netSEM structural equation models
    - take a look at SEM package

```
# install.packages("sem")
library(sem)
help(sem)


# install.packages("lavaan")
library(lavaan)
```

```
## This is lavaan 0.6-13
## lavaan is FREE software! Please report any bugs.
```

```
##
## Attaching package: 'lavaan'
```

```
## The following objects are masked from 'package:sem':
##
##     cfa, sem
```

```
help(lavaan)

# install.packages("netSEM")
library(netSEM)
help(netSEM)
```

2.2.2.4.2 Expected Values of a Predictive Model

- Now, once you have a predictive model

How well does it do, fitting your actual response?

- Remember a function is by definition single-valued
    - for a given value x_1 of the independent variable X
    - there is only dependent value y_1 for the dependent variable Y
- Therefore it can never actually predict
    - the exact observed value of the response
- this is why we keep the error term ϵ explicit

The Expected Value of a Regression Function

- Our regression function is $Y = f(\mathbf{X}) + \epsilon$
- Gives the Expected value of the response for $X = 4$

Notation for this is:

$$f(4) = E(Y|X = 4)$$

Or for our vector **X**

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

2.2.2.4.3 The ideal or optimal predictor of Y

- Minimizes the **loss function**
  - between the function and the data
- For example minimizing the sum of squared errors

2.2.2.4.4 An estimate (one version) of $f(X)$

- is called $\hat{f}(X)$
- since we could determine many versions of $f(X)$

And then we'll determine the best one of these $\hat{f}(X)$ functions

- That reduces the loss function

2.2.2.4.5 And then we are left with the irreducible error

- Which is just the variance of the errors.

2.2.2.4.6 So by better model building

- we can reduce the reducible error
- and we're left with the irreducible error.
  - Which I think of as the true "noise" in the data

2.2.2.5 Overview of the Regression Function and its nature

2.2.2.5.1 How do we estimate the function $f(X)$?

- We can perform the loss function minimization, at each specific value x of X.
  - Or at least in the neighborhood of x,
    * which is denoted by $\mathcal{N}(x)$
    * and called Nearest Neighbor Averaging

Note that the regression function $f(X)$ is not an algebraic function

- We didn't guesstimate it should be quadratic or some such.
- It is a numerical function defined for each value x of X

2.2.2.6 The Curse of Dimensionality

- When we are doing our nearest neighborhood averaging
  - in high dimensional datasets
  - we are hit by the curse of dimensionality
    * We can't define who are nearest neighbors
    * Because they tend to be far away in high dimensions

This hits us in many places of Prediction, Modeling and Statistical Learning

- The Curse of Dimensionality

The regression function $f(x)$

- Is also defined for vector X; e.g.
  $f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$
- Is the *ideal* or *optimal* predictor of Y with regard to mean-squared prediction error: $f(x) = E(Y|X = x)$ is the function that minimizes $E[(Y - g(X))^2|X = x]$ over all functions g at all points $X = x$.
- $\epsilon = Y - f(x)$ is the *irreducible* error — i.e. even if we knew $f(x)$, we would still make errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.
- For any estimate $\hat{f}(x)$ of $f(x)$, we have

$$E[(Y - \hat{f}(X))^2|X = x] = \underbrace{[f(x) - \hat{f}(x)]^2}_{\textit{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\textit{Irreducible}}$$

Figure 2: the regression function and its nature

How to estimate f

- Typically we have few if any data points with $X = 4$ exactly.
- So we cannot compute $E(Y|X = x)$!
- Relax the definition and let

$$\hat{f}(x) = \text{Ave}(Y|X \in \mathcal{N}(x))$$

where $\mathcal{N}(x)$ is some *neighborhood* of x.

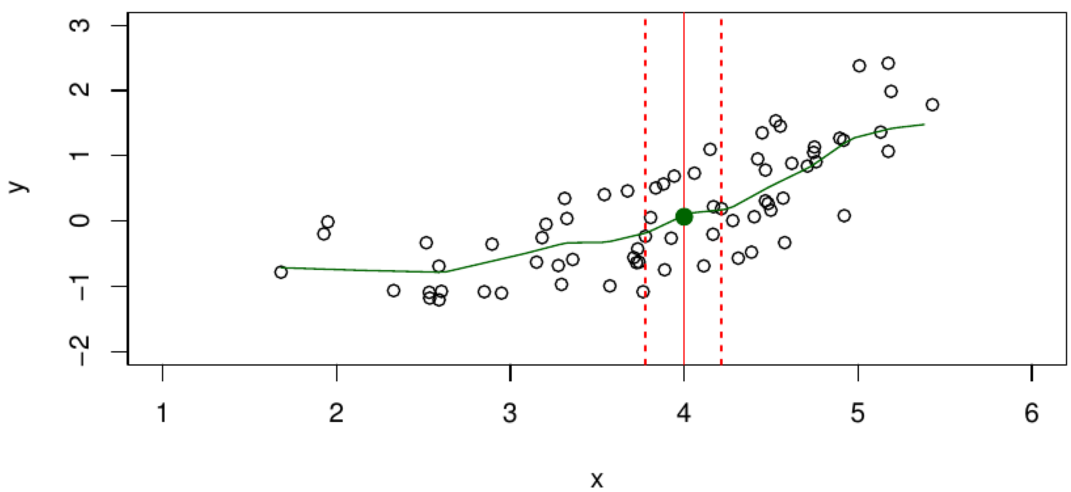


Figure 3: how to determine the regression function $f(X)$

2.2.2.7 Parametric and Structured Models

- One way to get around the curse of dimensionality,
  - Use Parametric Models

$$f_l(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ...\beta_p X_p$$

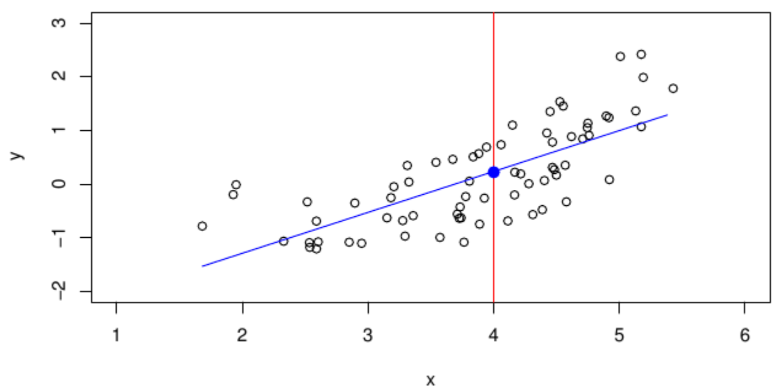
Where there are $p + 1$ parameters in the model

- Which are estimated by fitting the model to the data

Estimated values of a parameter β

- are denoted as $\hat{\beta}$

A linear model $\hat{f}_L(X) = \hat{\beta}_0 + \hat{\beta}_1 X$ gives a reasonable fit here



A quadratic model $\hat{f}_Q(X) = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$ fits slightly better.

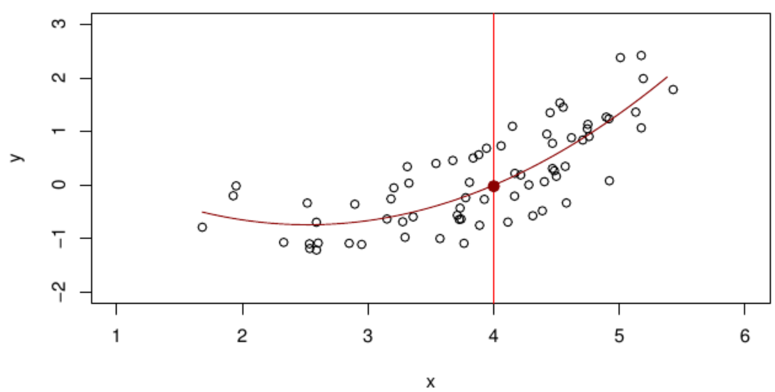


Figure 4: Examples of Parametric Models

2.2.2.7.1 Some tradeoffs in regression modeling

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret;
  - thin-plate splines are not.

- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often prefer a simpler model
    - involving fewer variables
  - Over a black-box predictor
    - involving them all.

2.2.2.7.2 Interpretability vs Flexibility

- Here are some of the approaches we'll look at this semester
  - Simpler models could be more interpretable
    - Or could be too naive
  - Flexibility makes for good fits
    - But can lead to overfitting

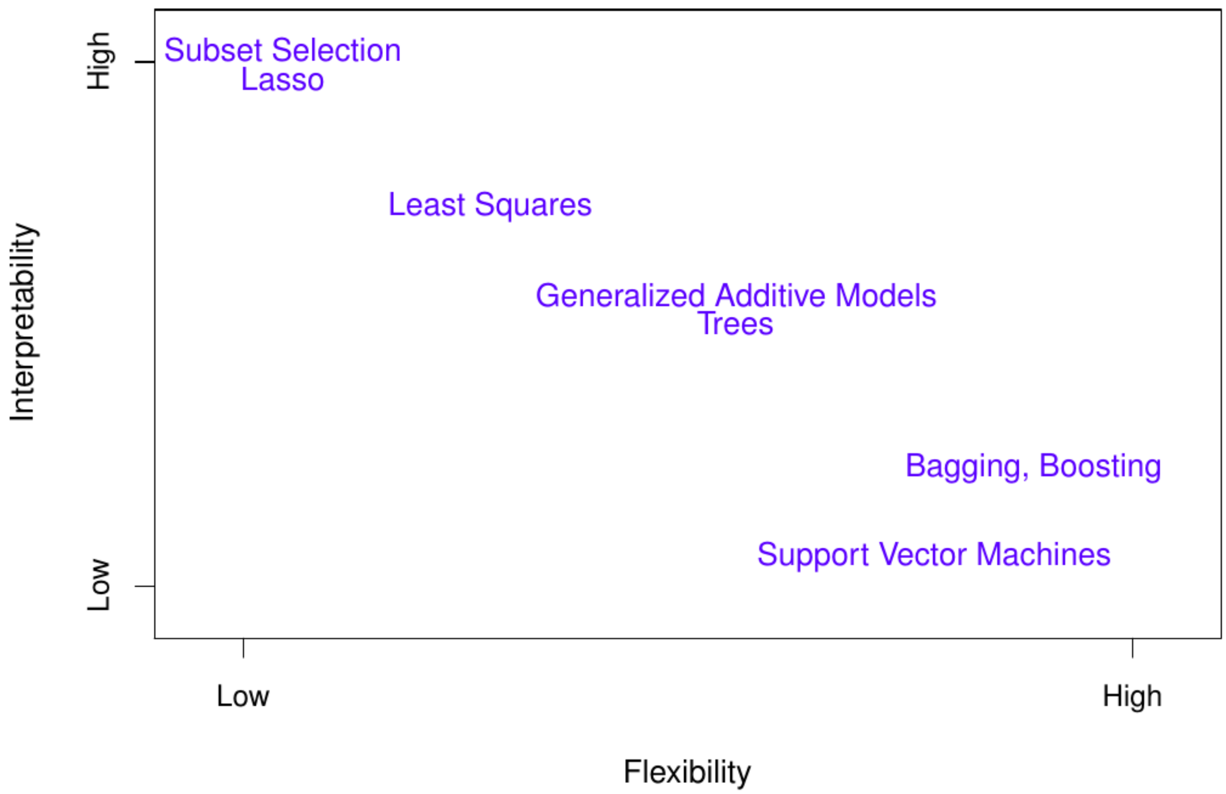


Figure 5: Interpretability vs Flexibility

2.2.2.8 Assessing Model Accuracy

2.2.2.8.1 Have to use training (Tr) and testing (Te) datasets

- To determine the best predictive model

2.2.2.9 The Bias vs. Variance Trade-off

- The hat is the estimated value of something. $\hat{f}(X)$

Assessing Model Accuracy

Suppose we fit a model $\hat{f}(x)$ to some training data $\mathsf{Tr} = \{x_i, y_i\}_1^N$, and we wish to see how well it performs.

- We could compute the average squared prediction error over Tr:

$$\mathrm{MSE}_{\mathsf{Tr}} = \mathrm{Ave}_{i \in \mathsf{Tr}}[y_i - \hat{f}(x_i)]^2$$

This may be biased toward more overfit models.

- Instead we should, if possible, compute it using fresh *test* data $\mathsf{Te} = \{x_i, y_i\}_1^M$:

$$\mathrm{MSE}_{\mathsf{Te}} = \mathrm{Ave}_{i \in \mathsf{Te}}[y_i - \hat{f}(x_i)]^2$$

Figure 6: Assessing Model Accuracy

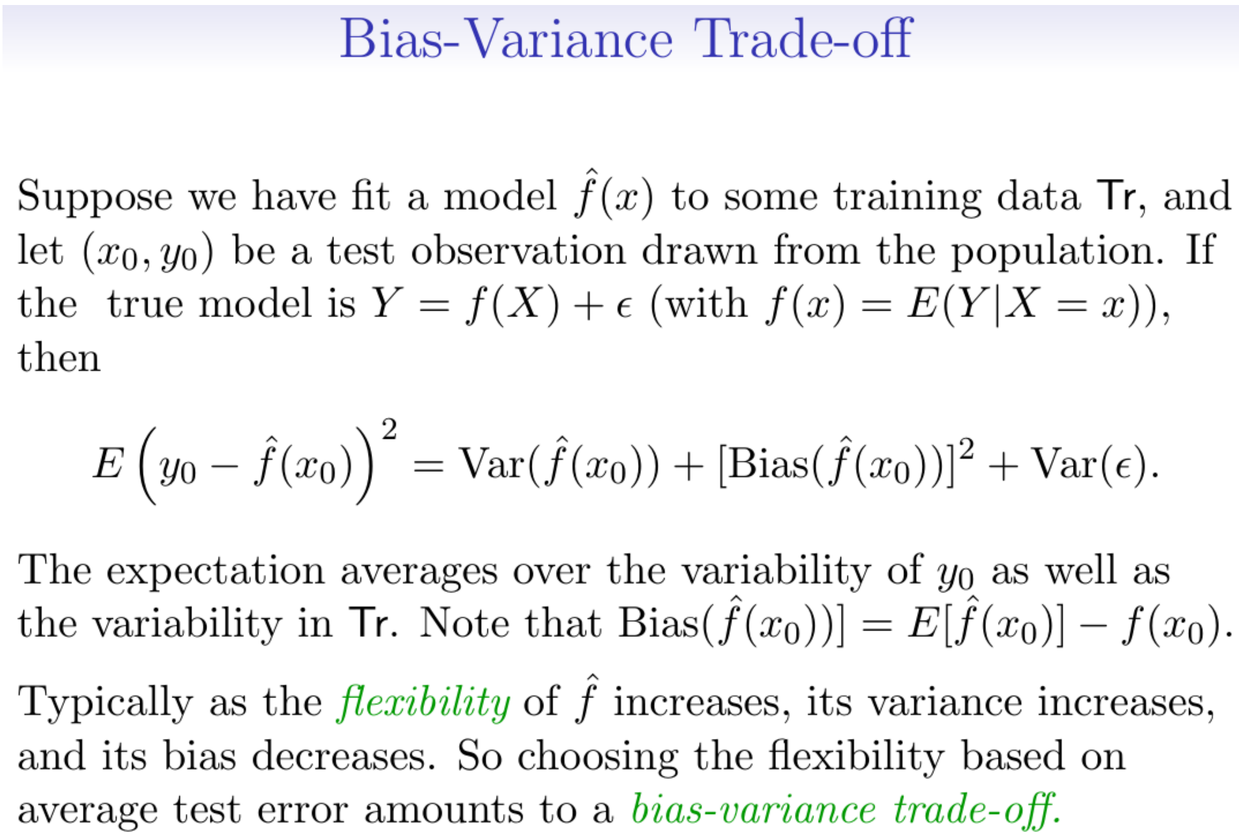


Figure 7: Bias vs. Variance Trade-off

- We can see the variance of $\hat{f}(X)$
- And the bias in $\hat{f}(X)$

Choosing the flexibility of your fitting function

- (i.e the number of predictors, or coefficients, in your model function)
- based on average test error
- amounts to what we call a bias-variance trade-off

And we use training datasets and testing datasets

- which we apply our model to
- to determine the optimal tradeoff we should use
- for a specific problem and model

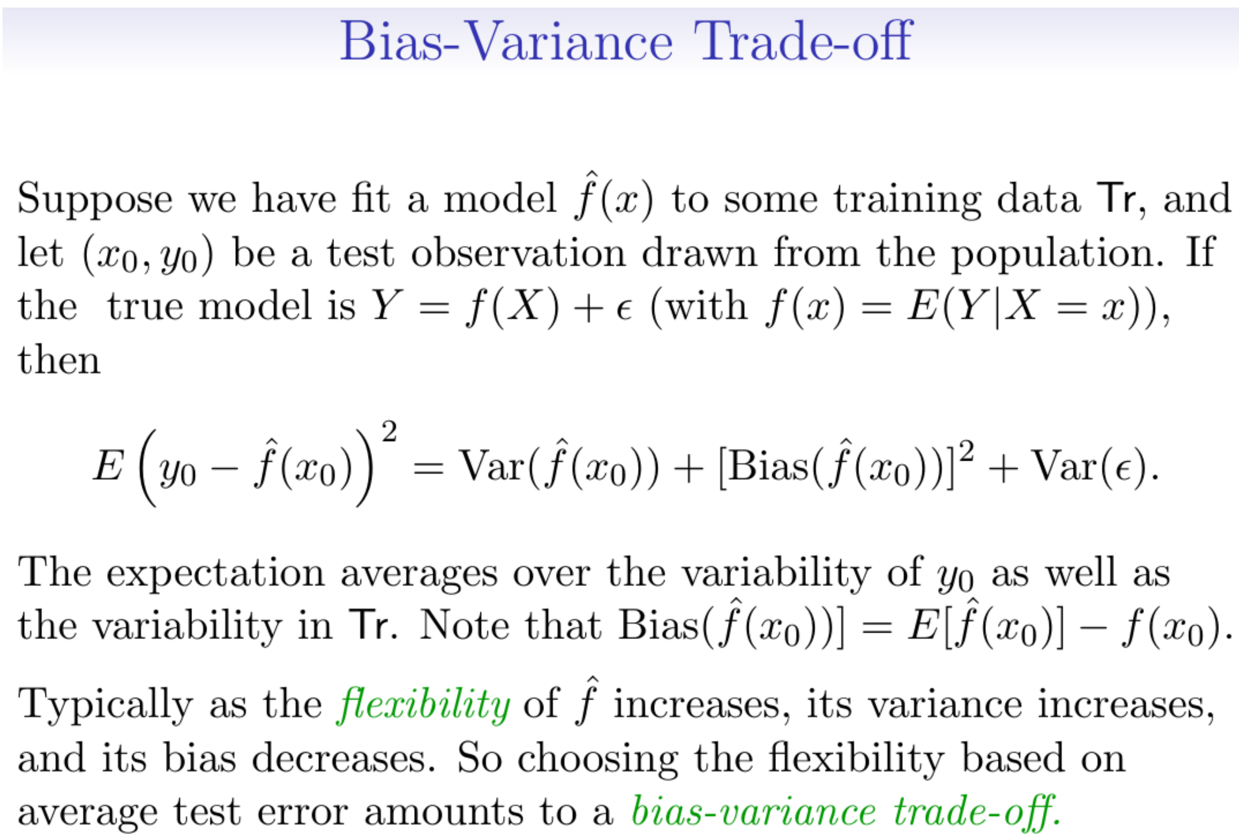


Figure 8: Bias vs. Variance in a Training & Testing Framework

2.2.2.9.1 How does all this play out in Classification Problems

- As opposed to Regression Problems, which we just discussed

2.2.2.10 Citations

- R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing, 2014..
- G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: 2nd Ed., with Applications in R, 2nd ed. 2021 edition. New York: Springer, 2021.
- Abbass Al Sharif. "Applied Modern Statistical Learning Techniques." [Abbass-Al-Sharif. Accessed January 17, 2016.(http://www.alsharif.info/).

- Diez, David M., Christopher D. Barr, and Mine Çetinkaya-Rundel. OpenIntro Statistics: Third Edition. 3 edition. S.l.: OpenIntro, Inc., 2015.
- Mayor, Eric. Learning Predictive Analytics with R. Packt Publishing - ebooks, 2015.