Lecture 2: Data and Problem Classes

STAT 1361/2360: Statistical Learning and Data Science

University of Pittsburgh
Prof. Lucas Mentch

Data and Methods

- Last time we talked about the "science" in data science and how the field relates to others
- Today we'll talk about the "data" in data science and the way in which we classify problems
  - Supervised vs Unsupervised learning
  - Machine and Statistical learning
  - Classification vs. regression vs. clustering
  - Parametric vs. semi-parametric vs. nonparametric statistics
- We'll end by talking about bias vs variance and underfitting/overfitting – concepts that will be crucial as we move forward in the course
- **Note:** ISLR Ch. 2 ends with a discussion on Bayes Classifiers and k-nearest neighbors – we'll cover these in Ch. 4

The Data in Data Science

(Usually) Observe data of the form:

$$\begin{array}{c} \\ n \text{ observations (rows)} \end{array} \left\{ \begin{array}{c} \begin{array}{cccccc} \mathrm{Y} & X_1 & X_2 & \cdots & X_p \end{array} \\ \begin{pmatrix} y_1 & x_{1,1} & x_{2,1} & \cdots & x_{p,1} \\ \vdots & \vdots & \vdots & & \vdots \\ y_n & x_{1,n} & x_{2,n} & \cdots & x_{p,n} \end{pmatrix} \\ \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{p+1 \text{ variables (columns)}} \end{array} \right.$$

- Y is called the *response* (dependent variable, output variable)
- $X_1, \ldots, X_p$ are called *features* (covariates, predictors, independent variables, input variables)
- We say that the data is of size $n \times p$ (n observations on p features)
- Common to write $X = (X_1, X_2, \ldots, X_p)$ as shorthand for all features and think of the data as n ordered pairs $(x, y)_1, \ldots, (x, y)_n$

Setup

In most situations, we think of

$$Y = f(X) + \epsilon$$

where

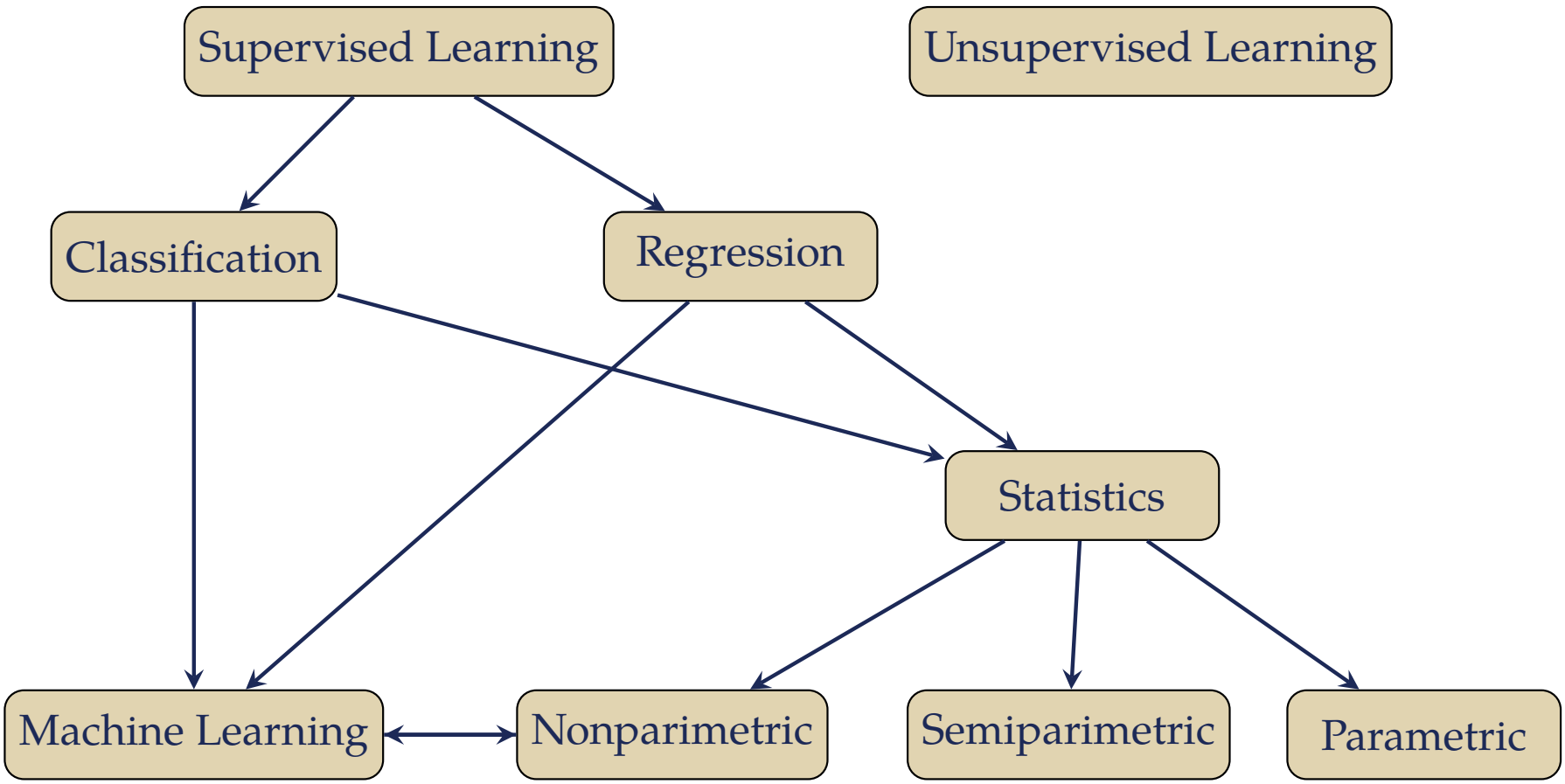
- f is some unknown function that describes the relationship between the features and response
- ϵ describes the (irreducible) error
  - Additional features X_{p+1} that are not measured
  - Measurement errors in recording the data

Supervised vs. Unsupervised

- Goal is to *learn* about the true underlying relationship f
  - The *learning* part of machine learning and statistical learning refers to *learning* something about relationships in the data
- Two* large classes of learning problems:
  1. **Supervised Learning:** As described above
  2. **Unsupervised Learning:** We observe features $X_1, \ldots, X_p$ but no response
     - Limited in what we can do in this situation; clustering
- We will focus almost exclusively on supervised learning problems in this course

*Semi-supervised learning (response is observed for some of the data points but not others); beyond the scope of this course

Classes of Problems



Classification vs Regression

- The kind of problem we undertake (classification or regression) depends heavily on the kind of response we have in the data
- **Categorical/Qualitative response:**

$$Y \in \{a_1, \ldots, a_m\}$$

e.g. Y measures ...

  - Letter grades on exam (A, B, C, D, F)
  - Presence/Absence of animal (1=present, 0=absent)
  - Quality score (good, better, best)

- **Continuous response:**

$$Y \in [a, b] \text{ or } \mathbb{R}$$

e.g. Y measures ...

  - Raw grades on exam (78.5, 82, 93, ...)
  - Salary/Wages/Revenue/Sales etc. ($25.5k, $67304, ...)
  - Quality score (97.8, 85.5, ...)

Categorical Data

- Problems with a categorical response are *usually* treated as **classification** problems (i.e. we want to predict/model a categorical response).

$$y_i \in \{a_1, ..., a_m\} \implies \hat{y}_i \in \{a_1, ..., a_m\} \text{ or } \hat{y}_i \in \mathbb{R}$$

- There are some notable exceptions to this, however. For example, if the data is *ordinal* (e.g. (0=poor, 1=satisfactory, 2=good, 3=great)) with many categories, regression might be a viable alternative.

- One common exception is with binary outcomes (0 =“Failure"; 1 = "Success"). In this case, averages of the responses have a natural interpretation as the “probability of success"

Continuous Data

- Problems with a continuous response are *usually* treated as **regression** problems (i.e. we want to predict/model a continuous response)

$$y_i \in \mathbb{R} \implies \hat{y}_i \in \mathbb{R}$$

- However, there can also be exceptions to this. Suppose for example that I measure car speeds along a certain stretch of road but am only interested in whether people are going over the speed limit:

$y_i \in \{46, 42, 54, 39, 47, ...\}$ but $\hat{y}_i \in \{\text{Over Limit, Under Limit}\}$

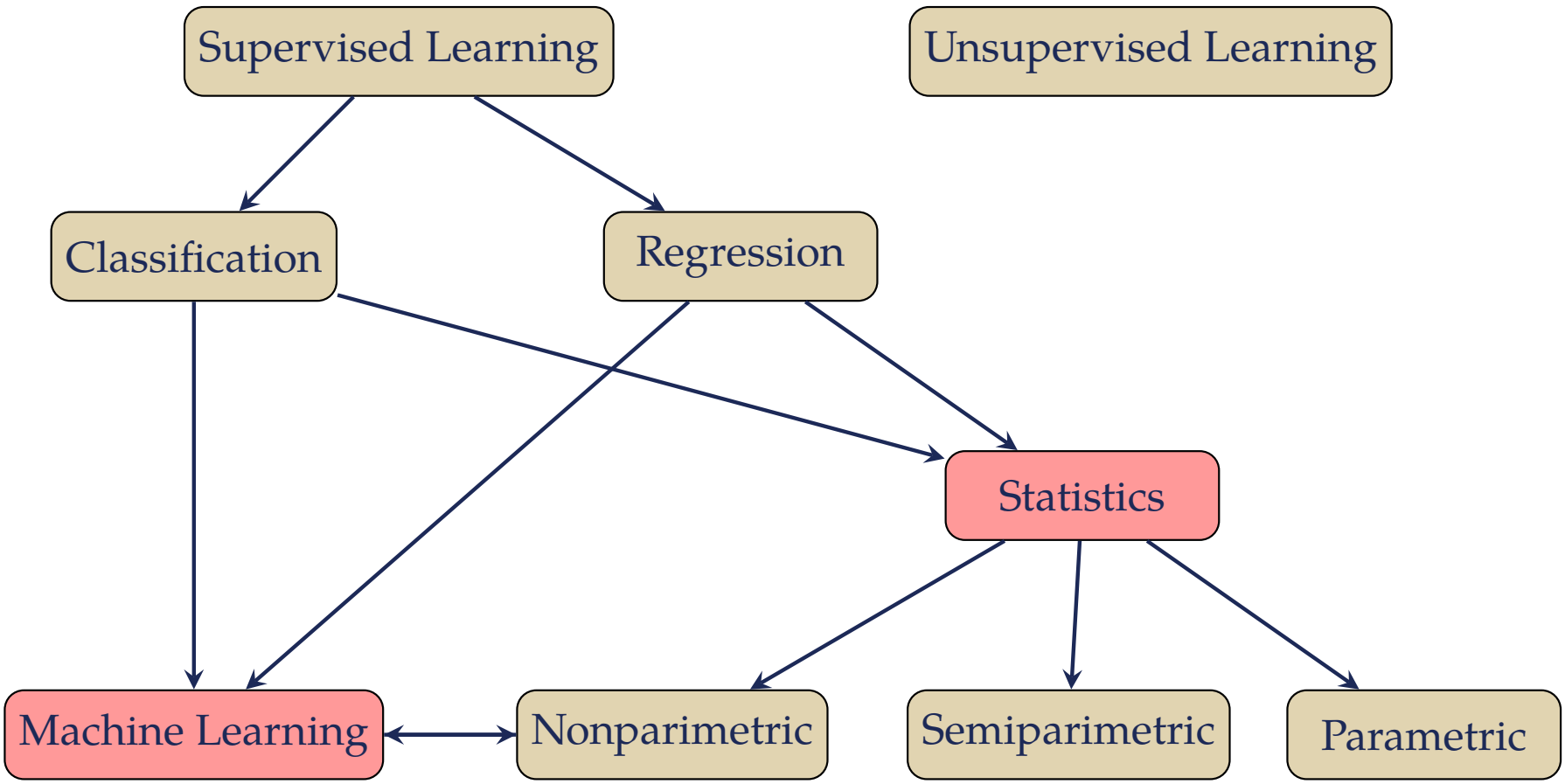
Takeaway

Takeaway: The type of response variable is obviously going to *play a big role* in determine whether we do classification or regression, but it does *not necessarily determine it*:

- It's possible to do regression even with a categorical response; it's possible to use a continuous response to ultimately perform some kind of classification
- What matters is what we're modeling/predicting – if we *predict/output* something categorical, we're doing classification. If we *predict/output* something continuous, we're doing regression.

Machine Learning vs Statistics

Classes of Problems



Setup

Recall that in supervised learning problems (regression *or* classification), we think of

$$Y = f(X) + \epsilon$$

where

- f is some unknown function that describes the relationship between the features and response
- ϵ denotes the (irreducible) error
- Y denotes the response variable
- $X = (X_1, \ldots, X_p)$ denotes the features that we believe Y depends on (or at least that we have data on)

Goals: Prediction

Now we need to ask: What kinds of questions do we want to be able to ask and answer?

There are usually two primary "classes" of goals:

1. **Prediction:** Given a feature vector $x^* = (x_1^*, \ldots, x_p^*)$, I want a prediction

$$\hat{y}^* = \hat{f}(x^*)$$

that is *as close to* the true (but unknown) response y^* as possible

   - I don't necessarily care anything about how f is modeled or what the form of the relationship between X and Y actually is, so long as I get good (accurate) predictions

Goals: Inference

2. **Inference:** I don't *not* care about predictive accuracy, but I want to know something about the underlying relationships in the data and be able to make some scientific inference

   - Does Y really depend on *every* predictor $X_1, ..., X_p$, or only a subset?
   - If I increase the value of some feature X_j, will that increase/decrease the value of Y?
   - What is the actual form the relationship between X and Y (e.g. is f linear)?
   - Which of the predictors seems to be "most important"?
   - Does a particular feature X_j add any explanatory power to the model? (i.e. could I get a model just as accurate without including X_j?)

Machine Learning vs Statistics

- In general, machine learning methods focus on prediction, statistical methods focus on inference:

$$\text{Machine Learning} \iff \text{Prediction}$$
$$\text{Statistics} \iff \text{Inference}$$

- Should **NOT** think of this as exclusively the way it is, but rather as the primary *goals* of the two approaches
  - Statistical models may sometimes generate very accurate predictions
  - Machine learning methods may sometimes have the ability to produce some kind of limited inference

$$Y = f(X) + \epsilon$$

- Goal is to construct $\hat{f}$ to be as close to f as possible
- We make little to no assumptions about the form of f; stitch together small-scale patterns to get $\hat{f}$
  - Very *flexible* model that is strongly driven by the data alone
  - Often results in algebraically complex form of $\hat{f}$;
  - *"Black-box"*: we may not even be able to write down a closed form for $\hat{f}$

Advantages:

- Given enough data (i.e. large enough n), predictions can be *very* accurate

Disadvantages:

- Often requires significant amount of data (minimum of hundreds to thousands)
- Limited ability to do inference
- Depending on the method, computing $\hat{f}$ may be computationally expensive

Statistics

- The statistical approach to these problems generally involves specifying/assuming a more explicit form of f. *How* explicit depends on the *kind* of statistics we're talking about:
  - **Parametric statistics/models/regression:** The exact form of f is specified up to a *finite* number of parameters; the data is then used to estimate those parmaters

    e.g. Linear Models:

    $$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

  - The problem of selecting $\hat{f}$ from all possible functions is greatly reduced to the much simpler problem of estimating $(\beta_0, \beta_1, ..., \beta_p)$.

- **Nonparametric statistics/models/regression:** f is assumed only to belong to some general (usually large) class of functions (e.g. f is twice-differentiable)
  - Specific $\hat{f}$ selected depends heavily on the data
  - Significant overlap with machine learning methods
- **Semiparametric statistics/models/regression:** f is assumed to consist of both a parametric and nonparametric portion:

$$\begin{aligned} \text{e.g.} \quad Y &= f(X) + \epsilon \\ &= h(X_1, \ldots, X_c) + g(X_{c+1}, \ldots, X_p) + \epsilon \\ &= \beta_0 + \beta_1 X_1 + \cdots + \beta_c X_c + g(X_{c+1}, \ldots, X_p) + \epsilon \end{aligned}$$

Advantages:

- Can often be fit with relatively small sample size n
- High potential for inference
- Typically easier to compute

Disadvantages:

"All models are wrong, but some are useful."

– George Box

- How much do I trust my inference if my model isn't "right"
- What if I have no idea what an appropriate model for f would look like?

Prediction vs Inference

Takeaway: Machine Learning vs Statistics; prediction vs inference

- There is always some **tradeoff** between predictive accuracy and inference/interpretability:
  - Given enough data, more flexible models will usually find a better $\hat{f}$, but may take a very complex form and not provide much use scientifically
  - On the other hand, if a strict form of f is assumed (think parametric statistics), we can more easily interpret the model and garner intuition for the underlying relationships

Assessing Model Fit

Overfitting

- What would be the downside to blindly throwing all of our data at a machine learning method?

Overfitting

- What would be the downside to blindly throwing all of our data at a machine learning method?
  - **Overfitting:**
    The $\hat{f}$ that we choose may be *very* accurate on the particular data that we observe, but may not *generalize* well to new data
  - We want predictions to be accurate on *all data*; given some new point $x^* = (x_1^*, \ldots, x_p^*)$, we want $\hat{y}^* = \hat{f}(x^*)$ to be accurate as well

Training vs Testing

- Standard solution is to split the n data samples into training and testing sets:

  Training Set: (70-80% of data) used to build/train the model (i.e. select $\hat{f}$)
  Test Set: (20-30% of data) used to assess how well $\hat{f}$ generalizes to new observations. Note that this data is *not* used in building $\hat{f}$

- Almost always, the splitting of the original sample into training and test sets is done completely at random

Goodness of fit

Assume for now that Y is continuous (or, at least that we're doing regression)

- How do we measure how well a particular model $\hat{f}$ actually fits the data we see?
- Some statistical models come with standard measures of goodness of fit; otherwise Mean Squared Error (MSE) is usually the default measure:

$$MSE = \frac{1}{n} \sum_{i=i}^{n} \left(y_i - \hat{f}(x_i) \right)^2$$

MSE

- MSE can be calculated on both the training and test sets:
  - **Training MSE** will be optimistically biased
  - **Test MSE** gives a better idea of generalizability

  (We'll talk about other ideas in ISLR Chapter 5)

- Goal is to choose $\hat{f}$ so as to minimize the MSE

How can we think about doing that?

MSE Decomposition

MSE can be decomposed as:

$$\mathbb{E}\left(y - \hat{f}(x)\right)^2 = \text{var}(\hat{f}(x)) + \text{bias}(\hat{f}(x))^2 + \underbrace{\text{var}(\epsilon)}_{\text{irreducible}}$$

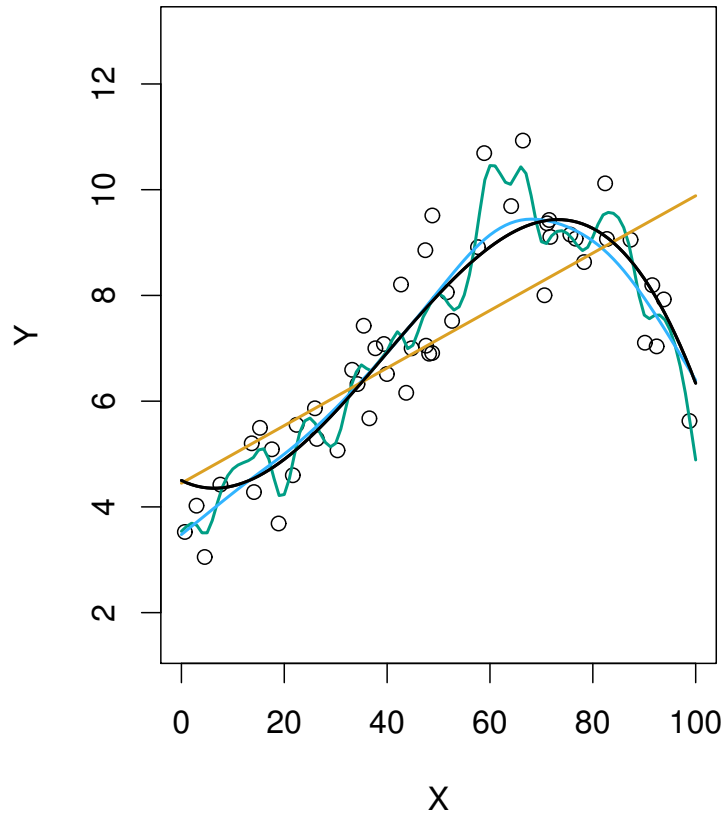
- $\text{var}(\hat{f}(x))$ = how much $\hat{f}(x)$ would change if reconstructed with new training data (not just different selection of training set from same original sample, but new data from a new sample)
- $\text{bias}(\hat{f}(x))$ = "true average error" between $f(x)$ and $\hat{f}(x)$; systematic error introduced by approximation

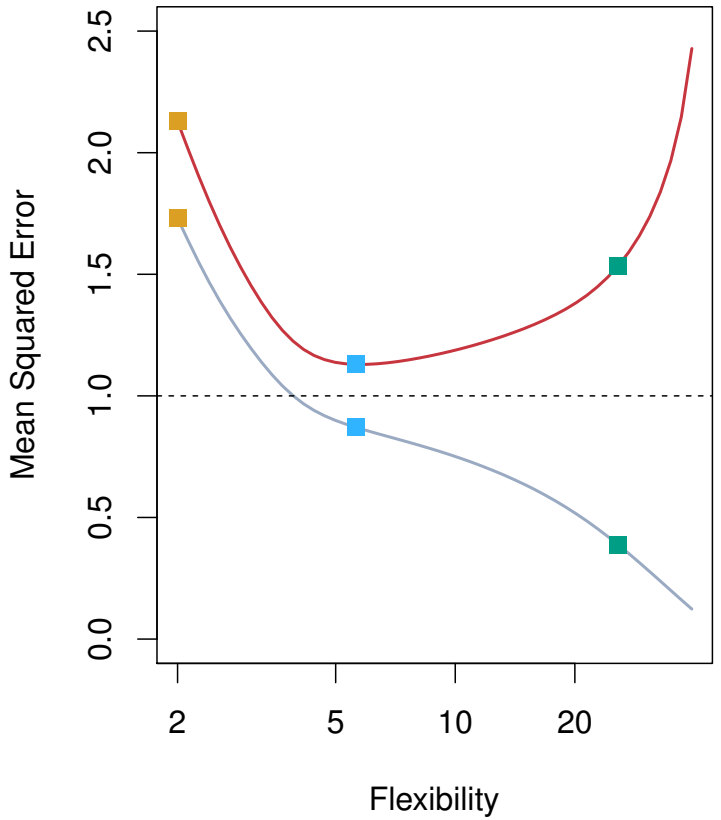
Bias-Variance Tradeoff

To minimize the MSE, we want to minimize both $\text{var}(\hat{f}(x))$ and $\text{bias}(\hat{f}(x))$, but there is a *tradeoff*:

- **Flexible models** are highly data-driven
  $\implies$ Usually can fit well to the data, but may get a very different model if trained with different training sample
  $\implies$ High Variance; Low Bias
- **More structured models** have very explicit form for $\hat{f}$
  $\implies$ Different training sets give very similar estimates of $\hat{f}$, but may *never* fit well if the imposed structure is much different than the true structure
  $\implies$ High Bias; Low Variance

Ex. 1: Bias-Variance Tradeoff





ISL Fig. 2.9 **Black:** = True model (f)

Gold: = Low flexibility $\hat{f}$ (low variance; high bias)

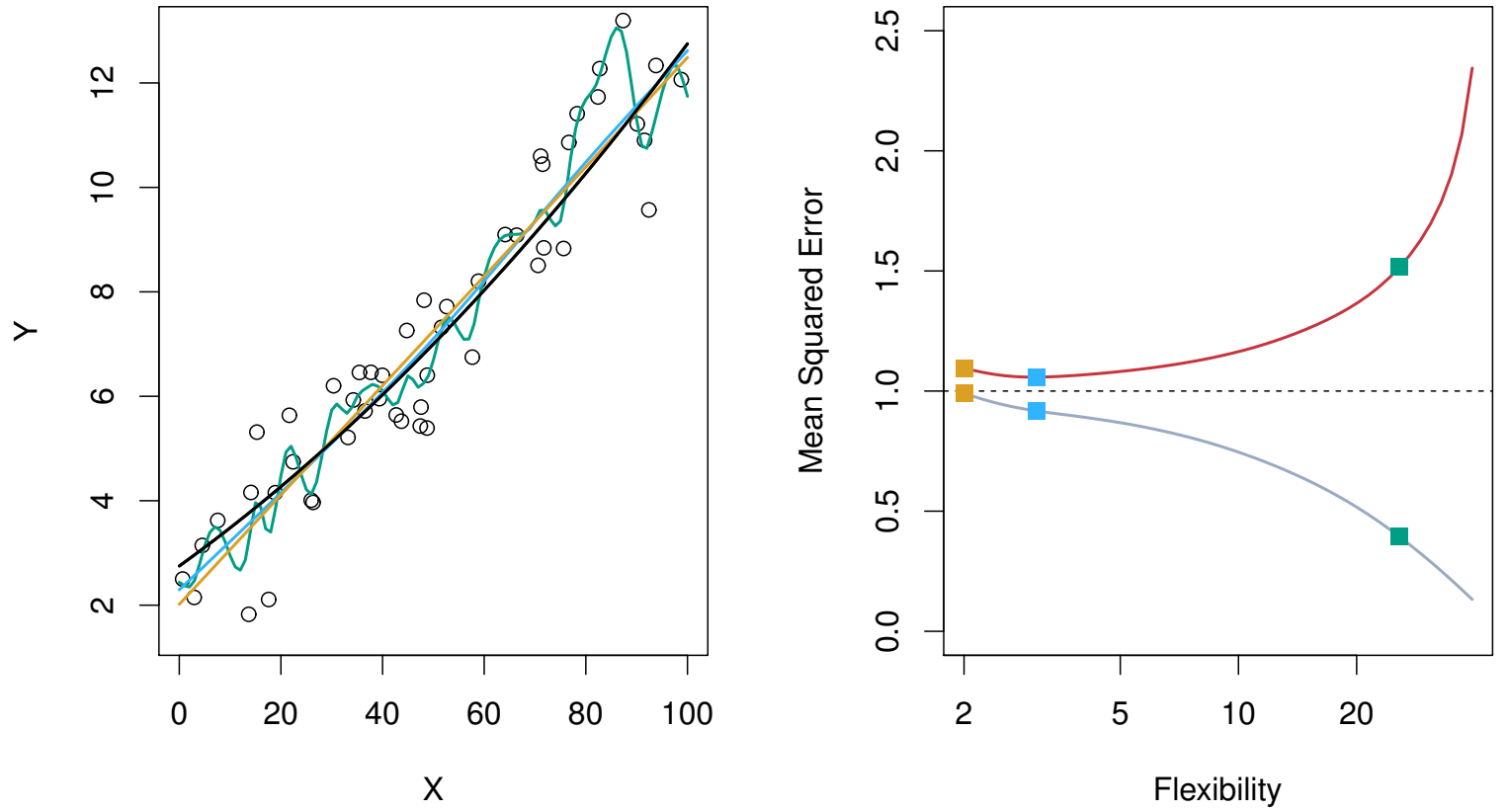
Blue: = Medium flexibility $\hat{f}$ (med variance; med bias)

Green: = High flexibility $\hat{f}$ (high variance; low bias)

Red Curve: = Test Error (MSE)

Gray Curve: = Training Error (MSE)

Ex. 2: Bias-Variance Tradeoff



ISL Fig. 2.10 **Black:** = True model (f)

Gold: = Low flexibility $\hat{f}$ (low variance; high bias)

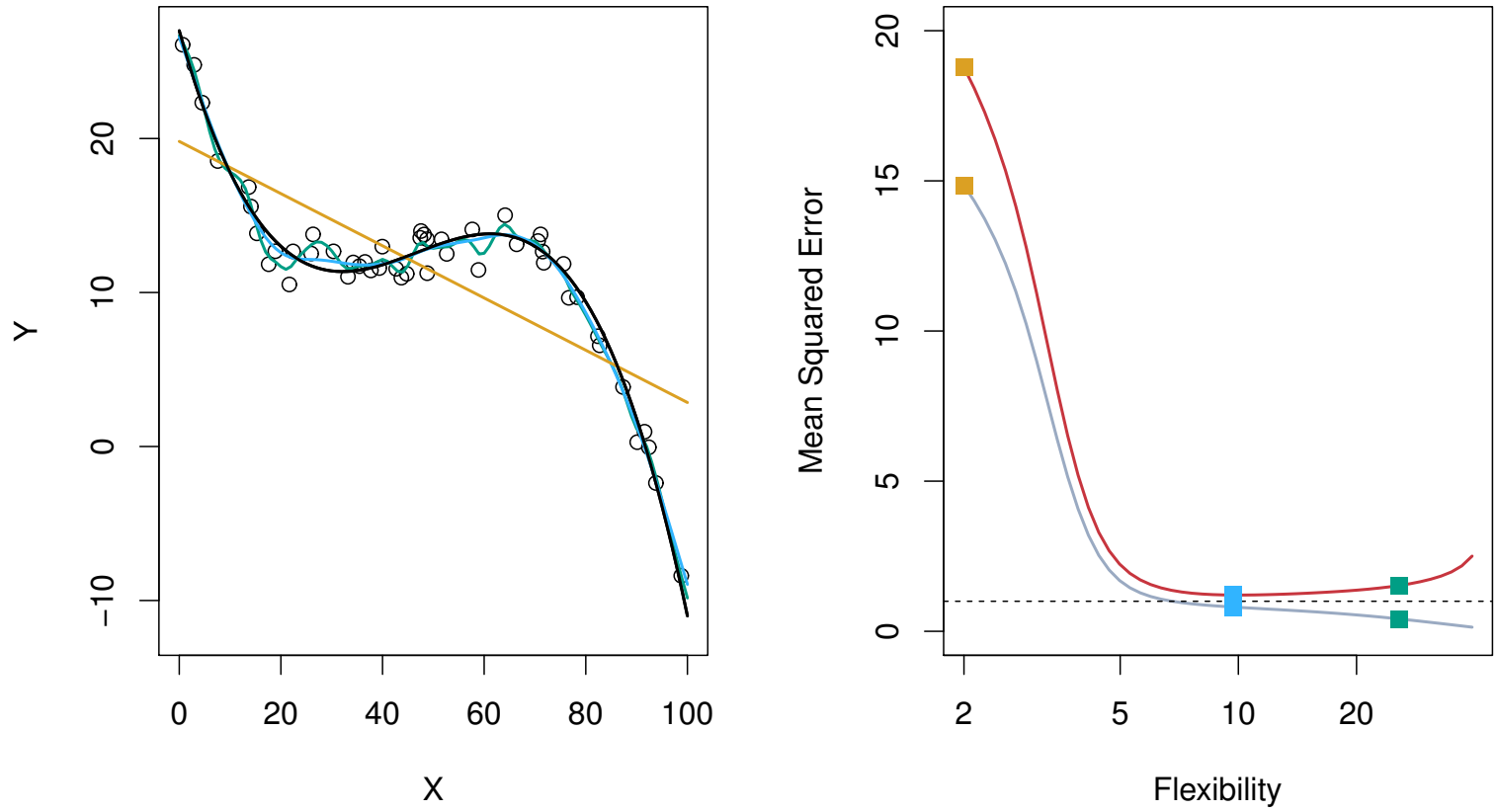
Blue: = Medium flexibility $\hat{f}$ (med variance; med bias)

Green: = High flexibility $\hat{f}$ (high variance; low bias)

Red Curve: = Test Error (MSE)

Gray Curve: = Training Error (MSE)

Ex. 3: Bias-Variance Tradeoff



ISL Fig. 2.11 **Black:** = True model (f)

Gold: = Low flexibility $\hat{f}$ (low variance; high bias)

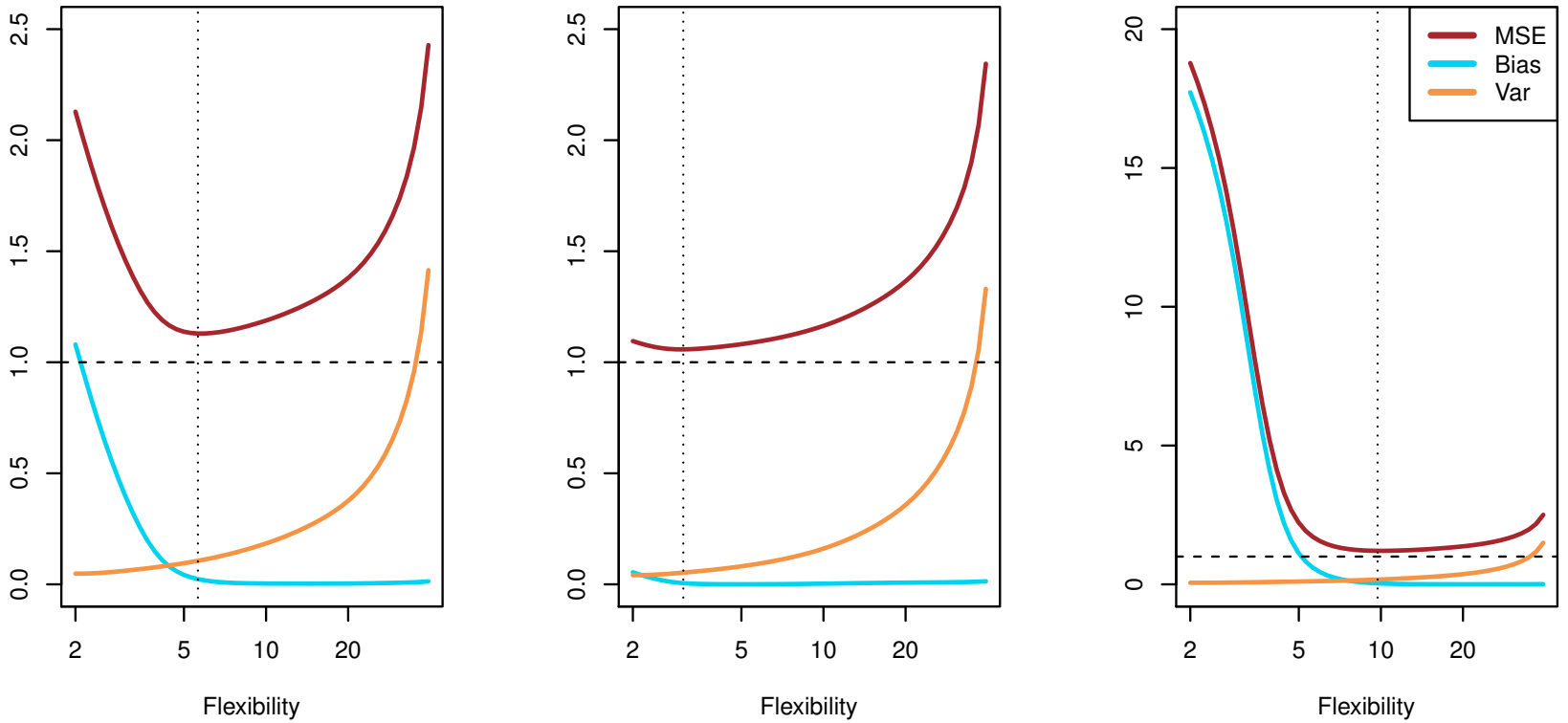
Blue: = Medium flexibility $\hat{f}$ (med variance; med bias)

Green: = High flexibility $\hat{f}$ (high variance; low bias)

Red Curve: = Test Error (MSE)

Gray Curve: = Training Error (MSE)

Bias-Variance Tradeoff



ISL Fig. 2.12: MSE (red) vs $\text{var}(\hat{f})$ (orange) vs. $\text{bias}(\hat{f})^2$(blue) vs $\text{var}(\epsilon)$(dashed)

Measuring Flexibility

Note: The plots on the previous slides plotted model "flexibility" on the x-axis. There are various ways one could measure this, but the most common is with "degrees of freedom" (dof or df):

$$df(\hat{f}) = df(\hat{y}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{cov}(y_i, \hat{y}_i)$$

We'll talk about this in more detail in a later lecture, but it should feel intuitive – we're measuring how much the predictions $\hat{y}$ depend on the response values y:

More dependence $\implies$ More df $\implies$ More flexible/jumpy model

Classification Error

What if the response is categorical? If the response is not ordinal, MSE makes no sense.

- Instead we look at the misclassification (error) rate:

$$\text{Err}_{\hat{f}} = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq \hat{y}_i\right) = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq \hat{f}(x_i)\right)$$

where

$$I(cond) = \begin{cases} 1 & \text{if condition is true} \\ 0 & \text{if condition is false} \end{cases}$$

Recap - Machine Learning

- **Machine learning methods** produce a very flexible $\hat{f}$ but can be very sensitive to the data at hand (high variance).
  - Given "enough" data, they will typically produce far more accurate predictions than more restrictive models.
  - More data will help stabilize the model (reduce variance), but does nothing to increase interpretability (and ability to do inference)

Recap - Statistics

- **Explicit Statistical Models** will usually produce a similar $\hat{f}$ across different training sets (low variance).
  - Can produce accurate predictions in some circumstances (if the model specified is “close")
  - Generally come equipped with standard inference procedures (e.g. confidence intervals, hypothesis tests).
  - If the model that you specify *a priori* is not correct, you will **always** have some bias that does not decrease, even with large datasets; best you can hope for is that $\hat{f}$ is a good approximation.

Takeaway

So where do you start in practice? The key questions:

1. How much data do I have?
   - **A lot?** Machine learning methods will almost certainly fit better
   - **Not a lot?** You'll need at least a partially explicit statistical model to get something reasonable

2. What are my (scientific) goals?
   - **Prediction?** Machine Learning, though if there are relatively few features, it may be possible to explicitly specify a "very close" statistical model
   - **Inference?** Statistics, though some machine learning techniques can produce limited forms of inference

Final Note

We've talked about *machine learning* and *statistics*; what about **statistical learning**?

- Not all that well-defined; to a degree, this is somewhat synonymous with machine learning, possibly with more of an emphasis on methods that produce at least a partially explicit $\hat{f}$

- Can think of this as statistics, but where some part of the model can be improved/tuned (i.e. learned) given more data; we'll see specific examples later