POLITECNICO
MILANO 1863

Machine Learning

- Statistical Machine Learning -

Matteo Matteucci, PhD (matteo.matteucci@polimi.it)
Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

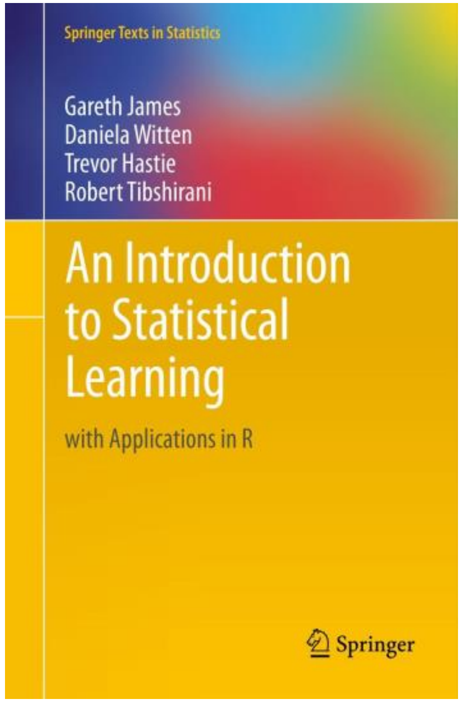
AIRLAB
ARTIFICIAL INTELLIGENCE AND ROBOTICS LAB

Reminder on Course Inspiration

Lectures are inspired by the book "An Introduction to Statistical Learning"

- Same authors of ESL, but ISL is easier!
- Practical perspective with labs and exercises using R language
- Available online as pdf (as ESL)

www.statlearning.com



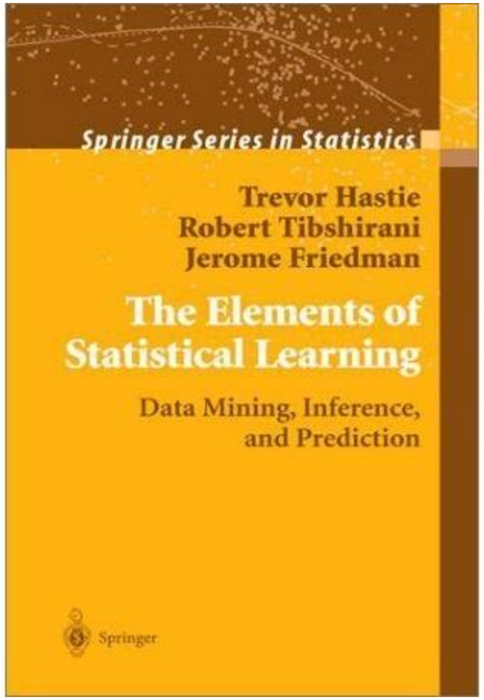
Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

Springer



Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Springer

Slides from the teacher (except for clustering) are taken from these books, while practicals have been rewritten from scratch … in python!

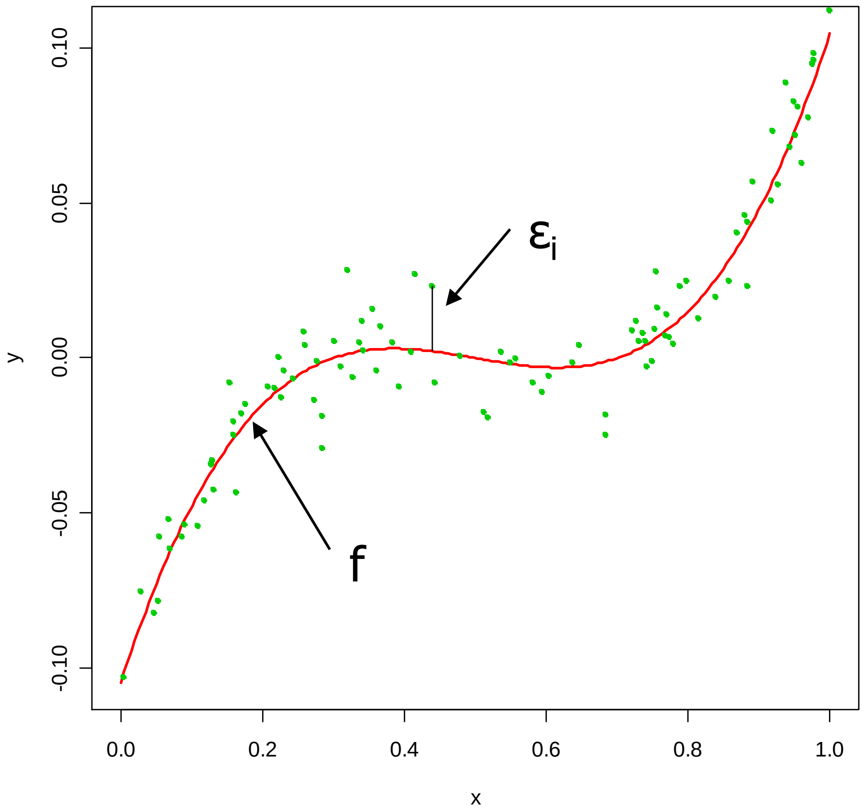
What is Statistical Learning?

Suppose we observe Y_i and $X_i = (X_{i1}, \dots, X_{ip})$ for $i = 1, \dots, N$

- Assume a relationship exists between Y and at least one of the observed X's
- Assume we can model this as

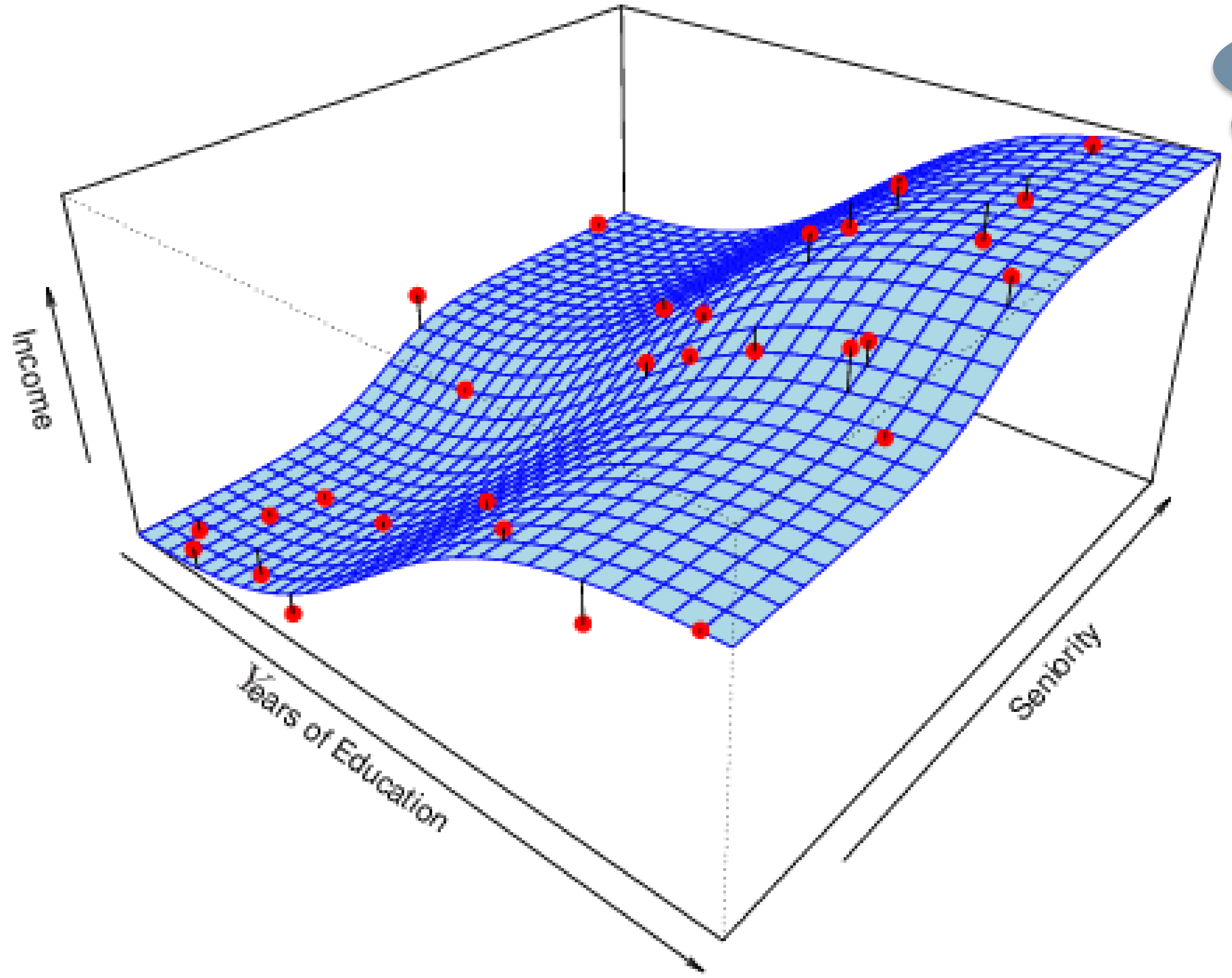
$$Y_i = f(X_i) + \varepsilon_i$$

  - *f* : unknown function systematic
  - ε_i : zero mean random error



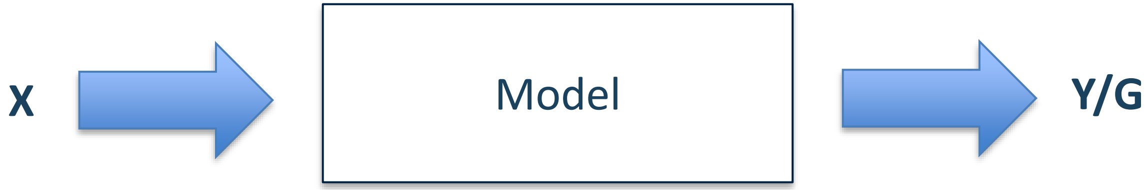
The term *Statistical Learning* refers to using the data to “learn” *f*

Example: Income vs. Education Seniority



Function f might also involve multiple variables …

Why do we estimate *f* ?



<u>Prediction:</u> Produce a good estimate for *f* to make accurate predictions for the response, Y/G, based on a new value of X.

<u>Inference:</u> Investigate the type of relationship between Y/G and the X's to control/influence Y/G.

- Which particular predictors actually affect the response?
- Is the relationship positive or negative?
- Is the relationship a simple linear one or is it more complicated etc.?

Examples for Prediction & Inference

Direct Mail Prediction

- Predicting how much money an individual will donate based on observations from 90,000 people on which we have recorded 400 different characteristics.
- Don’t care too much about each individual characteristic.
- Just want to know: For a given individual should I send out a mailing?

Medium House Price

- Which factors have the biggest effect on the response
- How big the effect is
- Want to know: how much impact does a river view have on the house value

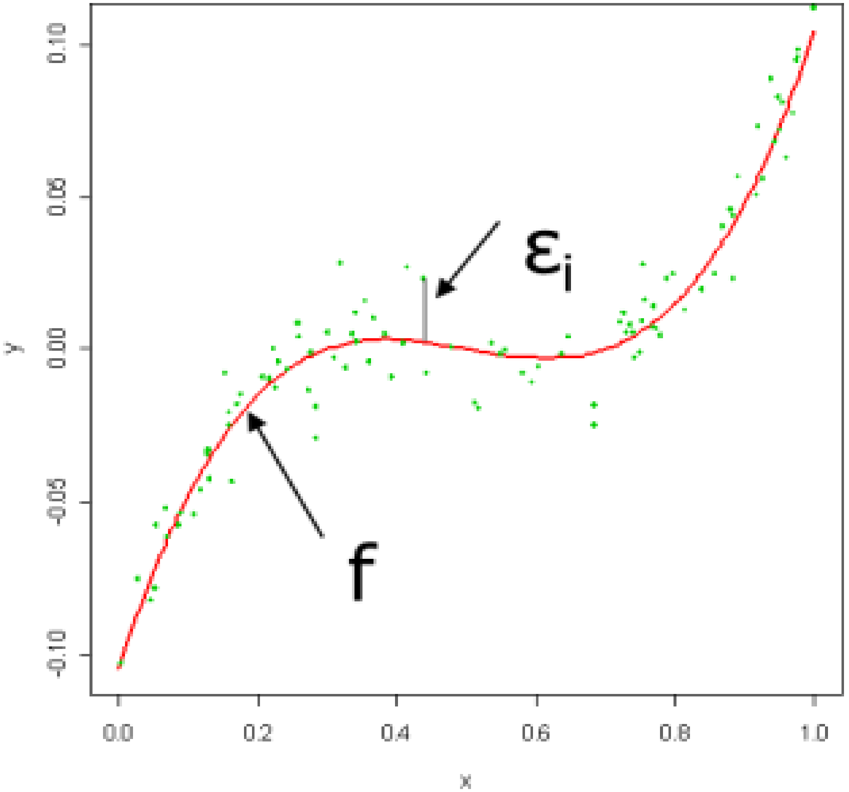
How Do We Estimate *f* ?

We have observed a set of *training data*

$$\{(\boldsymbol{X_1}, Y_1), (\boldsymbol{X_2}, Y_2), \ldots, (\boldsymbol{X_N}, Y_N)\}$$

Use statistical method/model to estimate *f* so that for any $(\boldsymbol{X_i}, Y_i)$

$$Y_i \approx \hat{f}(\boldsymbol{X_i})$$



Based on the model *f*, statistical methods/models are usually divided in

- Parametric Methods/Models
- Non-parametric Methods/Models

Parametric Methods (Part 1)

Parametric methods make an assumption about the model underlining *f*

- Reduce the problem of estimating *f* to estimating a set of parameters
- They involve a two-step model-based approach

STEP 1: Make some assumption about the functi
up with a model (e.g., a linear model)

We will see more flexible/powerful models than linear ones …

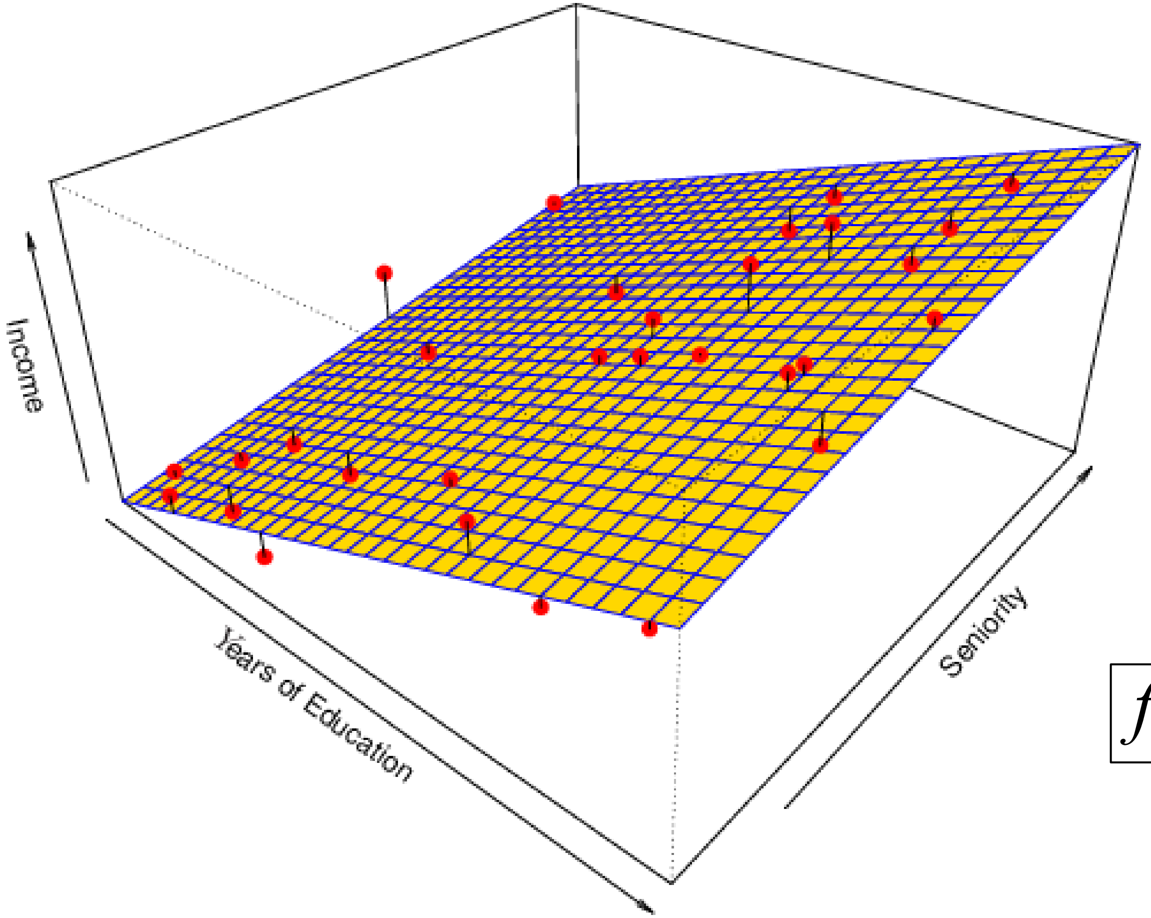
$$f(\mathbf{X}_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip}$$

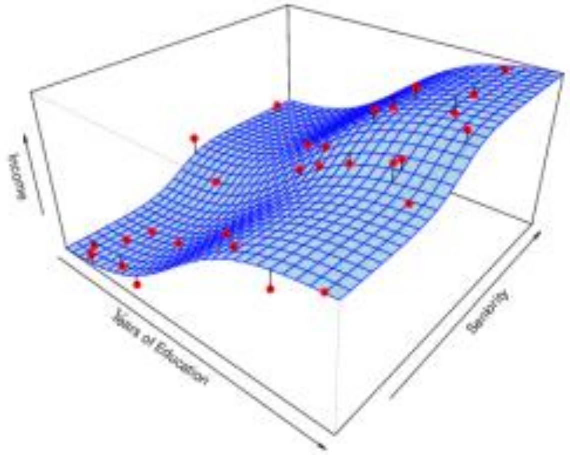
STEP 2: Use the training data to fit the model, i
unknown parameters

$$\beta_0 \quad \beta_1 \quad \beta_2 \quad \ldots \quad \beta_p$$

Ordinary Least Sqares are used for this, but alternative methods exists too.

Example: A Linear Regression Estimate





$$\boxed{f = b_0 + b_1 \,´\, Education + b_2 \,´\, Seniority}$$

Even if the standard deviation is low we will still get a bad answer if we use the wrong model (high bias).

Non-parametric Methods

Sometimes are referred as “sample-based” or “instance-based” methods, they do not make explicit assumptions about the functional form of *f*, and exploit the training data “directly”

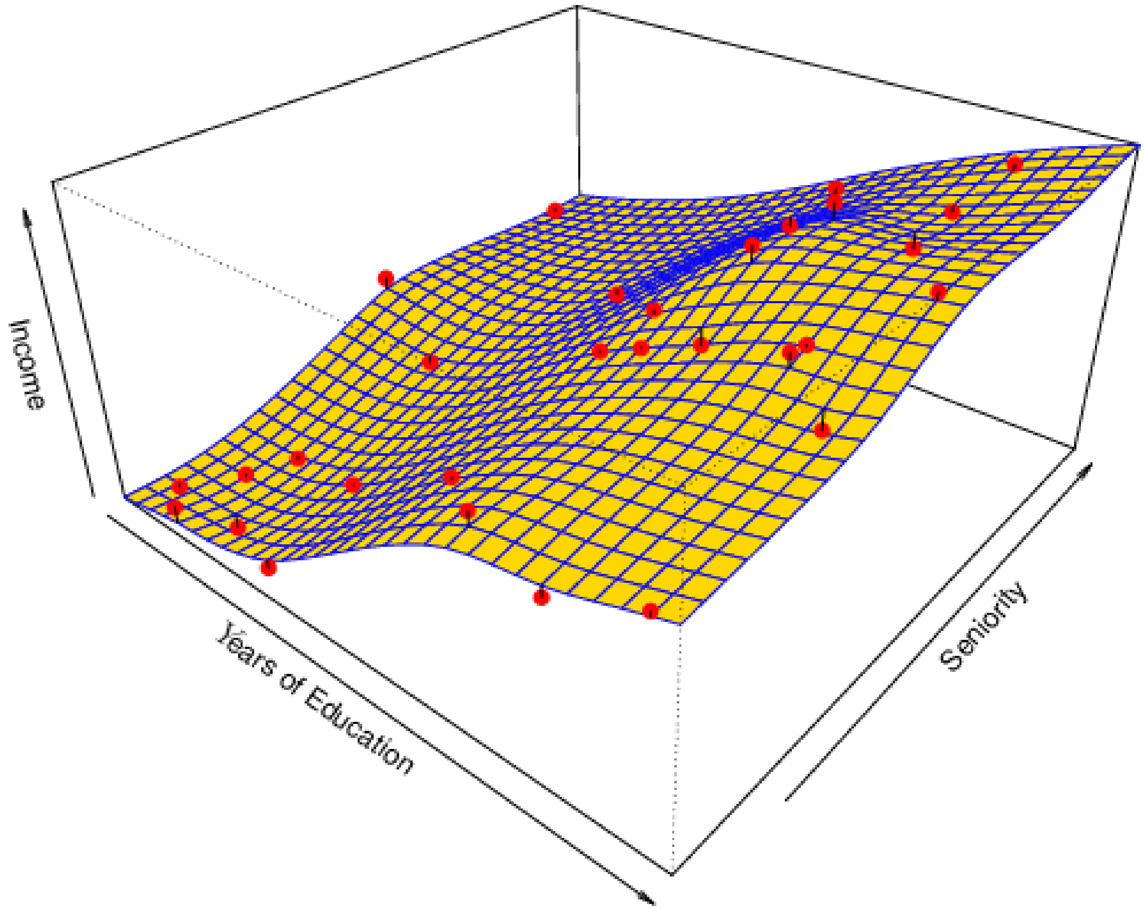
Advantages:

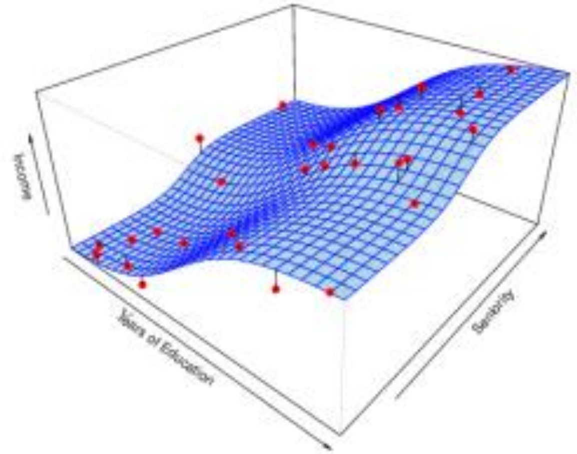
- They accurately fit a wider range of possible shapes of *f*
- They do not require a “training” phase

Disadvantages:

- A very large number of observations required to obtain an accurate estimate
- Higher computational cost at “testing” time
- They accurately fit a wider range of possible shapes of f.

Example: A Thin-Plate Spline Estimate





Smooth thin-plate spline fit

Non-parametric regression methods are more flexible thus they can potentially provide more accurate estimates

Prediction Accuracy vs Model Interpretability

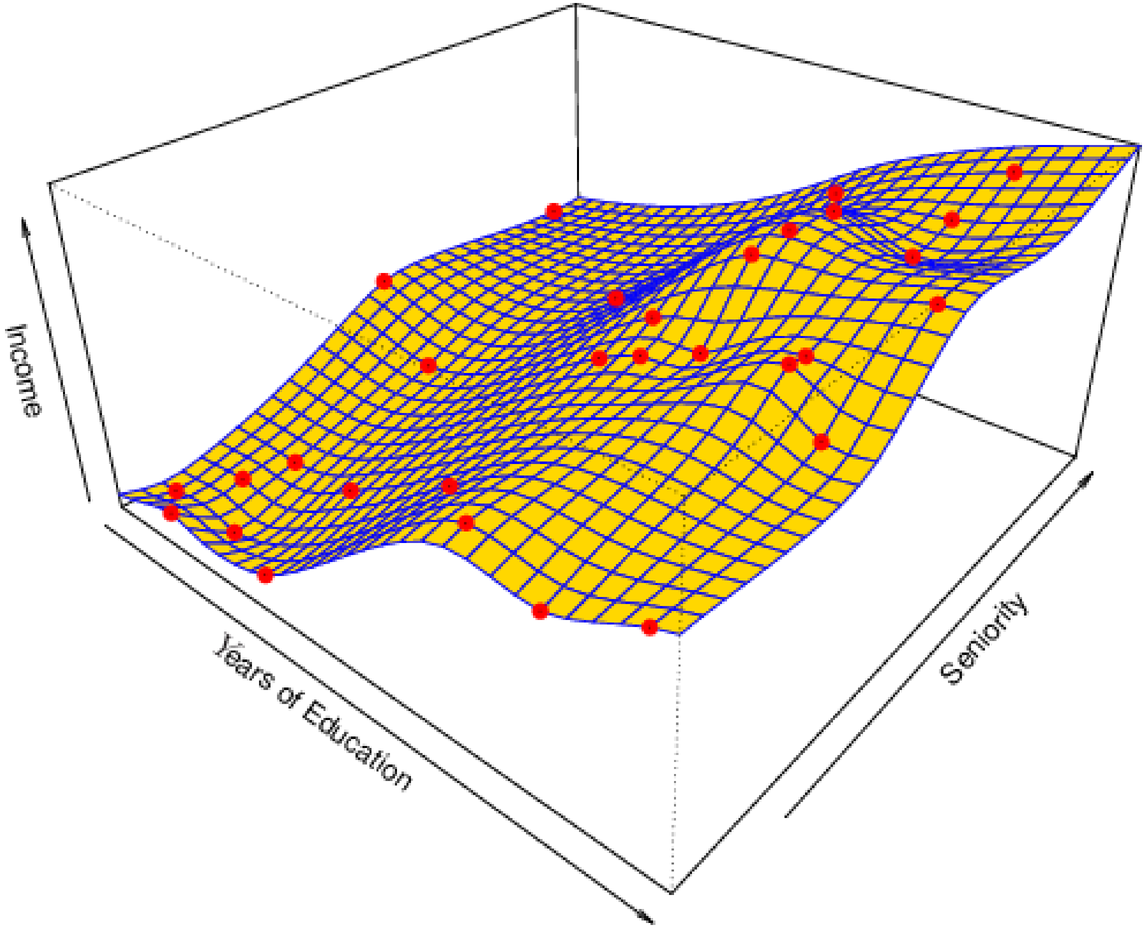
Why not just use a more flexible method if it is more realistic?

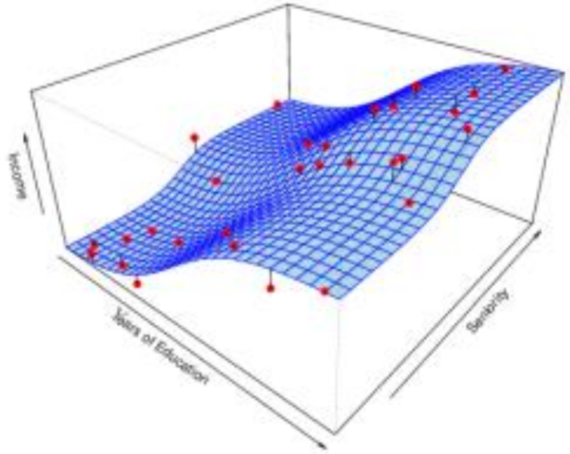
Reason 1: A simple method, e.g., linear regression, produces a model which is much easier to interpret (the Inference part is better).

- E.g., in a linear model, β_j is the average increase in Y for a one unit increase in X_j holding all other variables constant.

Reason 2: Even if interested in prediction, it is often possible to get more accurate predictions with a simple, instead of a complicated, model.

Example: A Poor Estimate

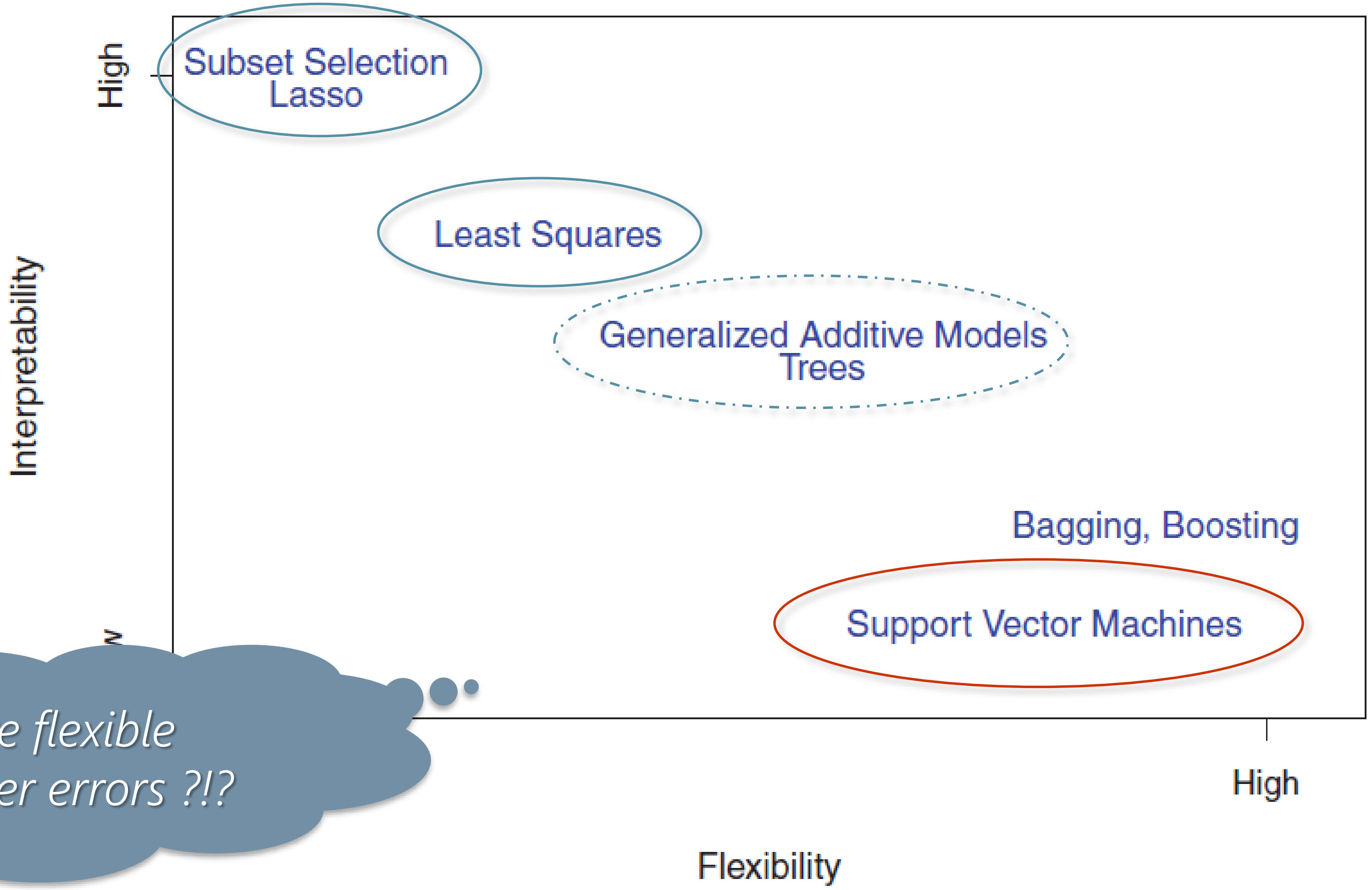




Thin-plate spline fit with zero training error

Non-parametric regression methods can also be too flexible and produce poor estimates for *f* (high variance)

Flexibility vs Model Interpretability



But more flexible means lower errors ?!?

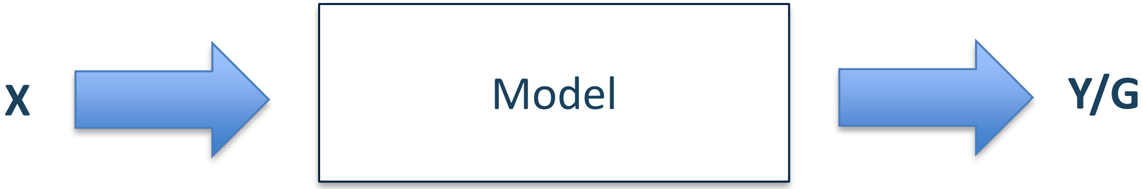
FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Reducible vs Irreducible Error

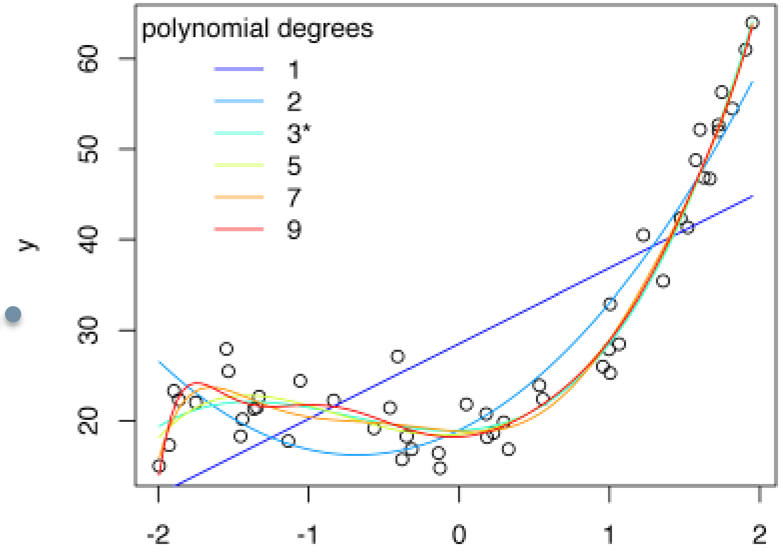
The error our estimate will have has two components

$$Y_i = f(X_i) + \varepsilon_i$$

- *Reducible error* due to the choice of *f (model complexity)*

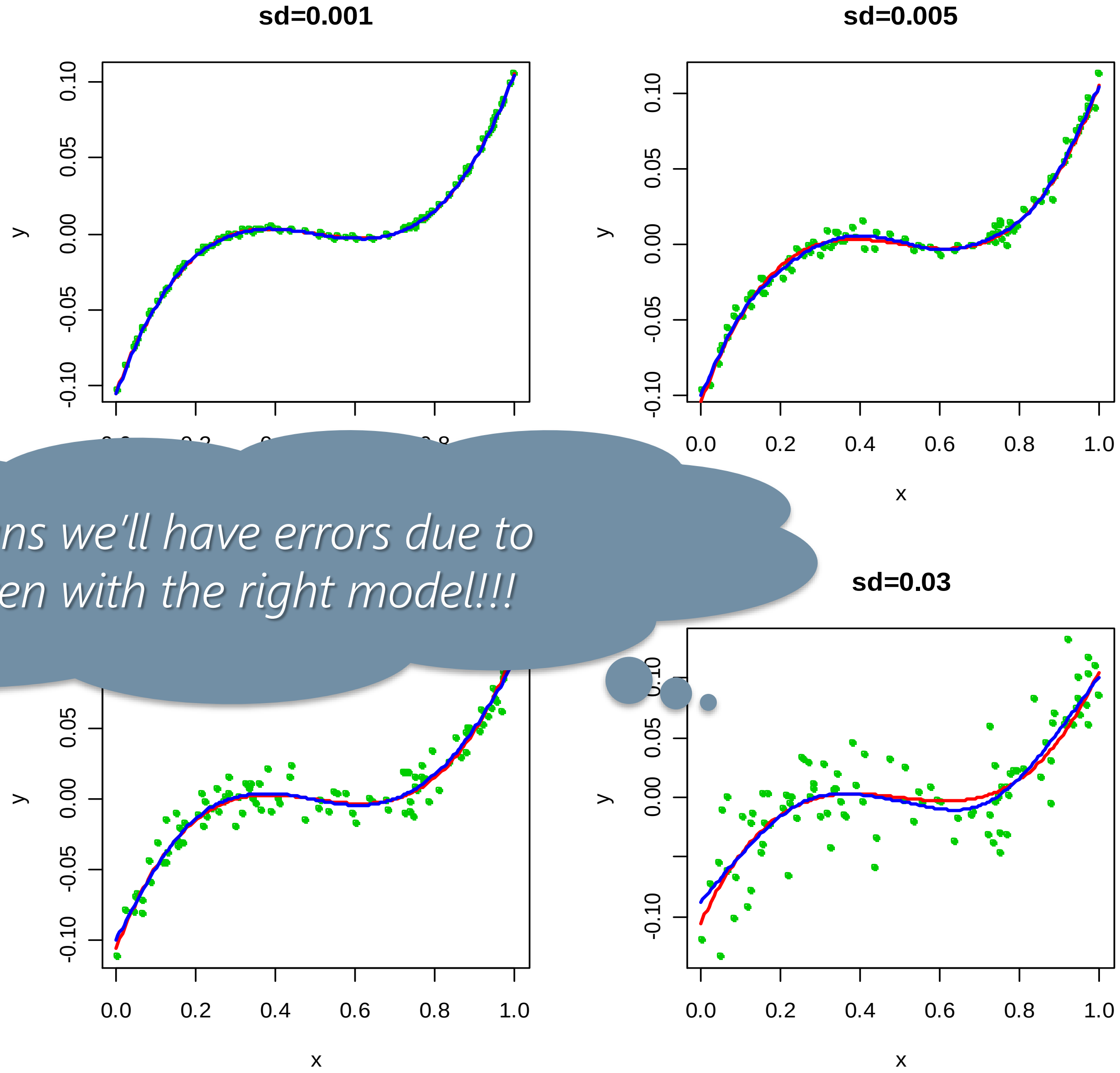


I will come back to this soon … several times!



- *Irreducible error* due to the presence of ε_i in the training set

Irreducible error … because noise matters!



This means we'll have errors due to noise even with the right model!!!

Reducible vs Irreducible Error (Part 2)

The error our estimate will have has two components

$$Y_i = f(X_i) + \varepsilon_i$$

- *Reducible error* due to the choice of *f (model complexity)*
- *Irreducible error* due to the presence of ε_i in the training set

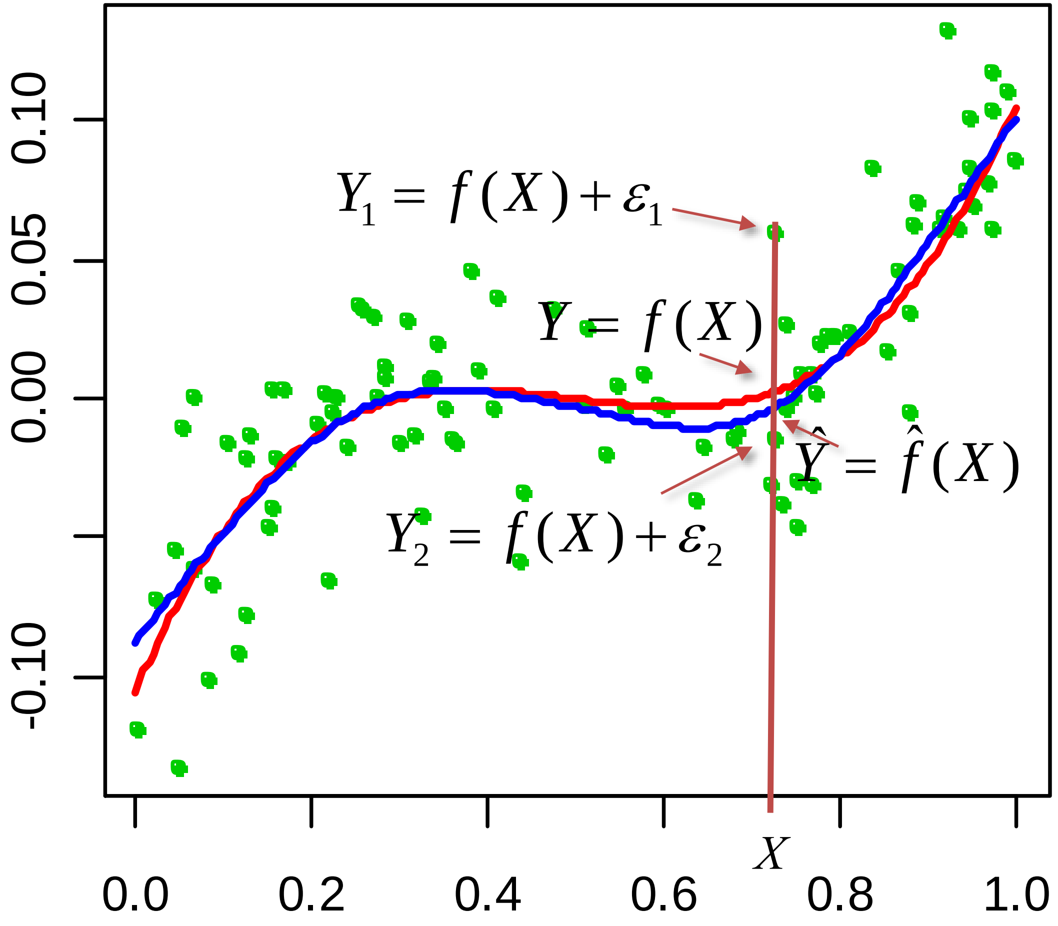
Let assume $\hat{f}$ and X fixed for the time being

Can you derive this?

$$\hat{Y} = \hat{f}(X)$$

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}} \end{aligned}$$

Reducible vs Irreducible Error (Part 3)



$$E[(Y - \hat{Y})^2] =$$

$$= E[(f(X) + \varepsilon - \hat{f}(X))^2] =$$

$$= E[f(X)^2 + \varepsilon^2 + \hat{f}(X)^2 - 2 \cdot \varepsilon \cdot f(X) - 2 \cdot \varepsilon \cdot \hat{f}(X) - 2 \cdot f(X) \cdot \hat{f}(X)] =$$

$$= f(X)^2 + E[\varepsilon^2] + \hat{f}(X)^2 + 2 \cdot E[\varepsilon] \cdot f(X) - 2 \cdot E[\varepsilon] \cdot \hat{f}(X) - 2 \cdot f(X) \cdot \hat{f}(X) =$$

$$= f(X)^2 + E[\varepsilon^2] + \hat{f}(X)^2 - 2 \cdot f(X) \cdot \hat{f}(X) =$$

$$= (f(X)^2 + \hat{f}(X)^2 - 2 \cdot f(X) \cdot \hat{f}(X)) + E[\varepsilon^2] =$$

$$= (f(X) - \hat{f}(X))^2 + E[\varepsilon^2] - 0 =$$

$$= (f(X) - \hat{f}(X))^2 + Var(\varepsilon)$$

Quality of Fit

Suppose we have a regression problem

- A common accuracy measure is mean squared error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Where $\hat{y}_i$ is the prediction for the observation in our training data.

Training is designed to make MSE small on training data, but …

- What we really care about is how well the method works on new data. We call this new data "**Test Data**".
- There is no guarantee that the method with the smallest *<u>Training MSE</u>* will have the smallest *<u>Test MSE</u>*

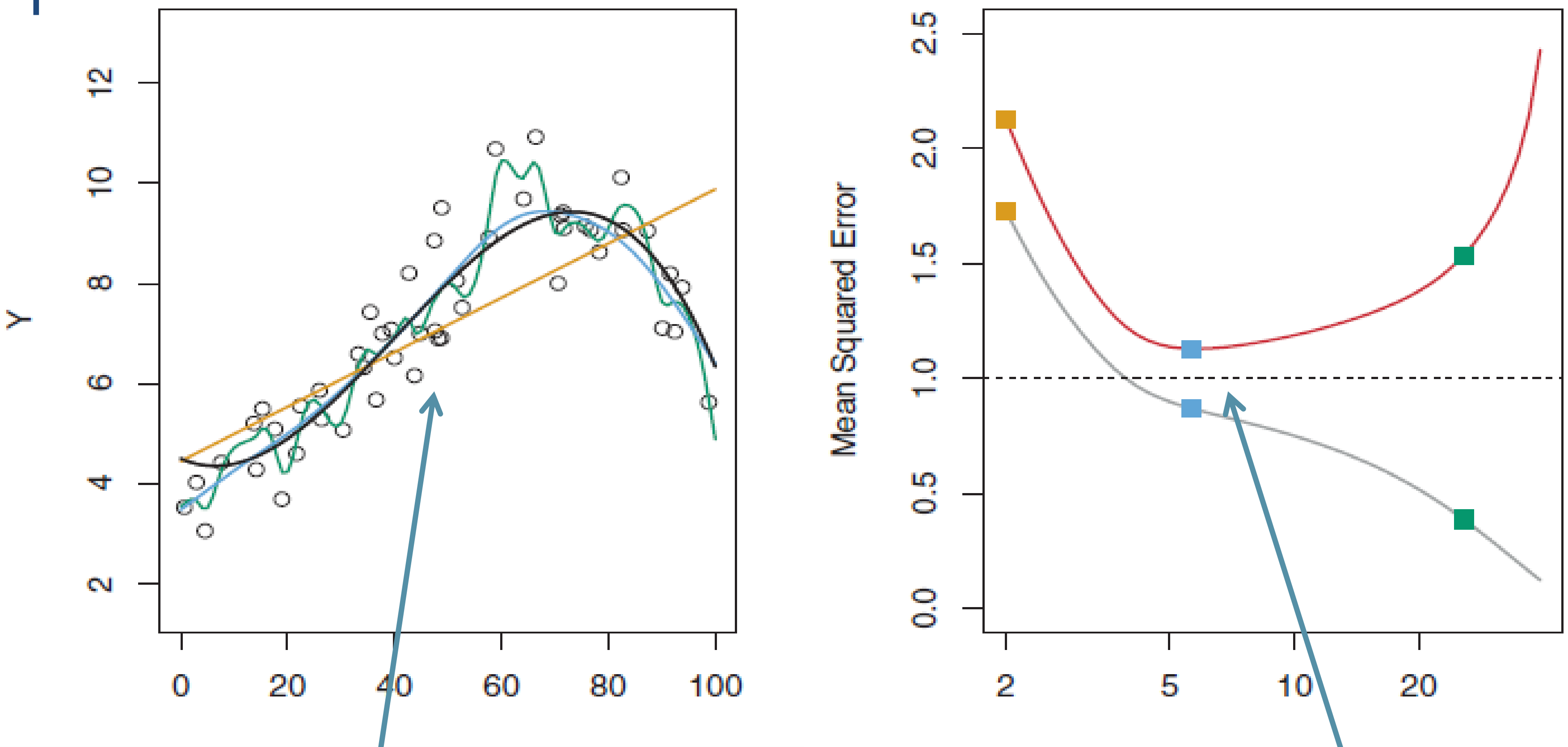
Training vs. Test Mean Squared Error

The more flexible a method is, the lower its training MSE will be, i.e., it will "fit" or explain the training data very well.

- *Side Note*: More Flexible methods (such as splines) can generate a wider range of possible shapes to estimate *f* as compared to less flexible and more restrictive methods (such as linear regression). The less flexible the method, the easier to interpret the model. Thus, there is a trade-off between flexibility and model interpretability.

However, the test MSE may in fact be higher for a more flexible method than for a simple approach like linear regression

Example 1

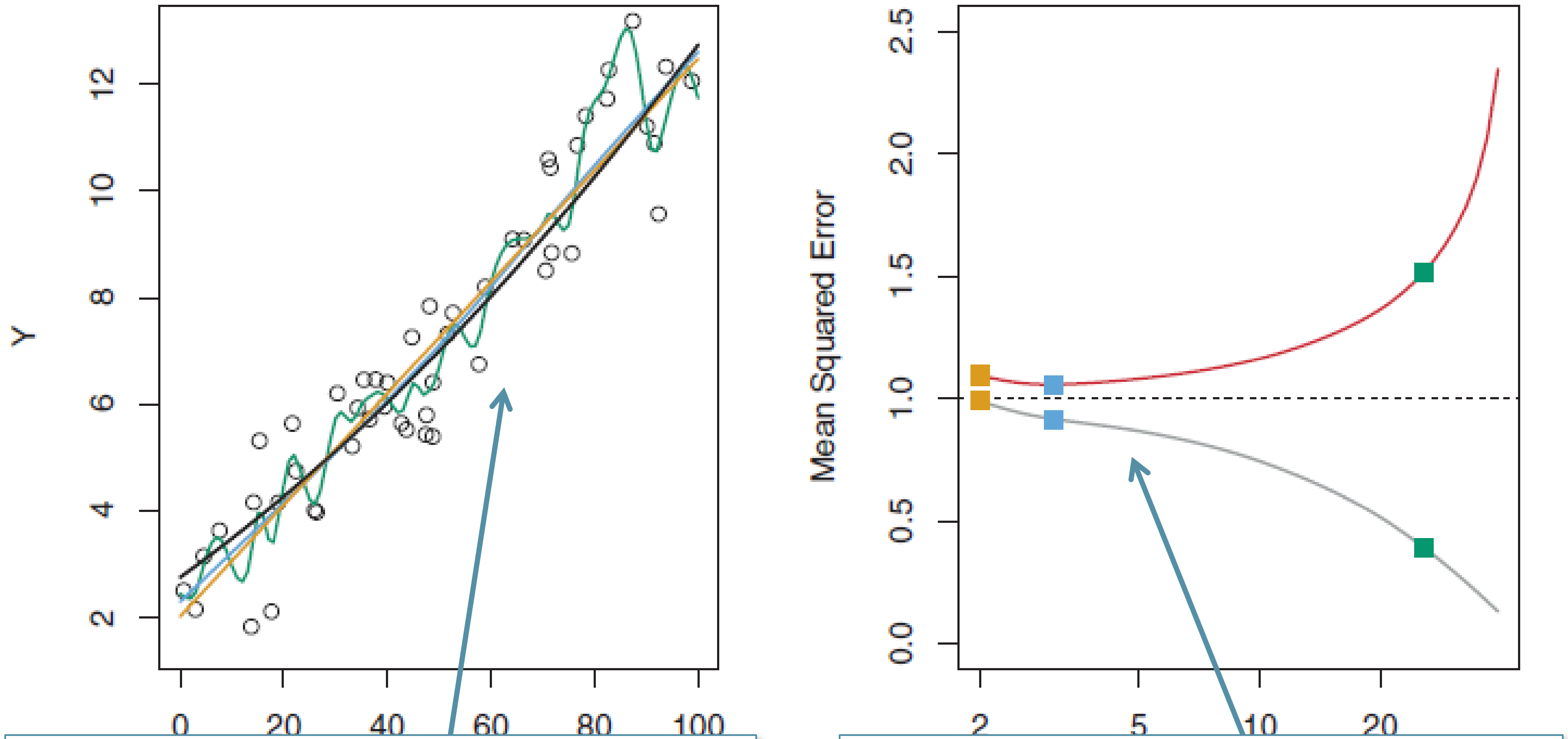


Black: Truth
Orange: Linear Estimate
Blue: smoothing spline
Green: smoothing spline

RED: Test MES
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)

curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Example 2

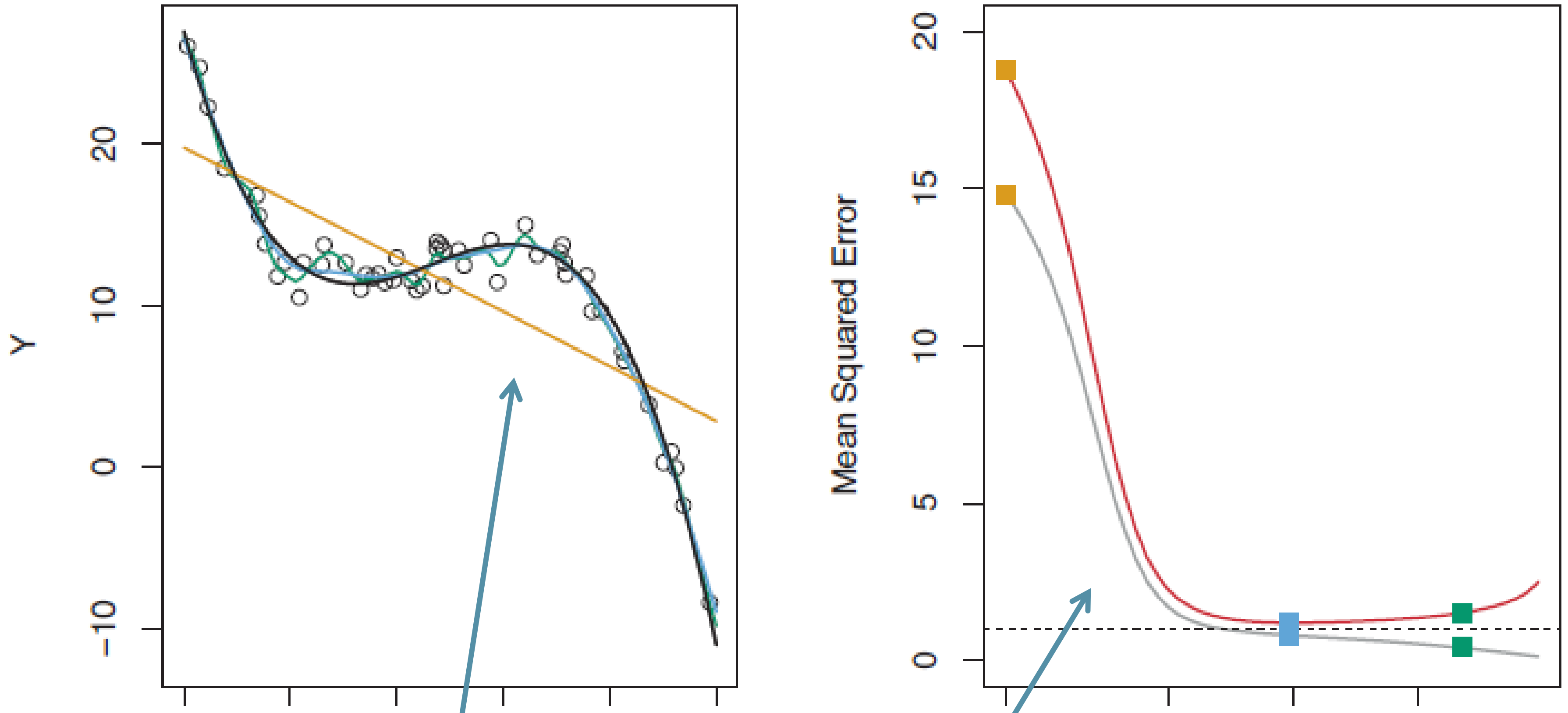


Black: Truth
Orange: Linear Estimate
Blue: smoothing spline
Green: smoothing spline

RED: Test MES
Grey: Training MSE
Dashed: Minimum possible test MSE
(irreducible error)

FI ure
much closer to linear. In this setting, linear regression produces a very good fit to
the data.

Example 3



Black: Truth
Orange: Linear Estimate
Blue: smoothing spline
Green: smoothing spline

RED: Test MES
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)

linear. In this setting, linear regression provides a very poor fit to the data.

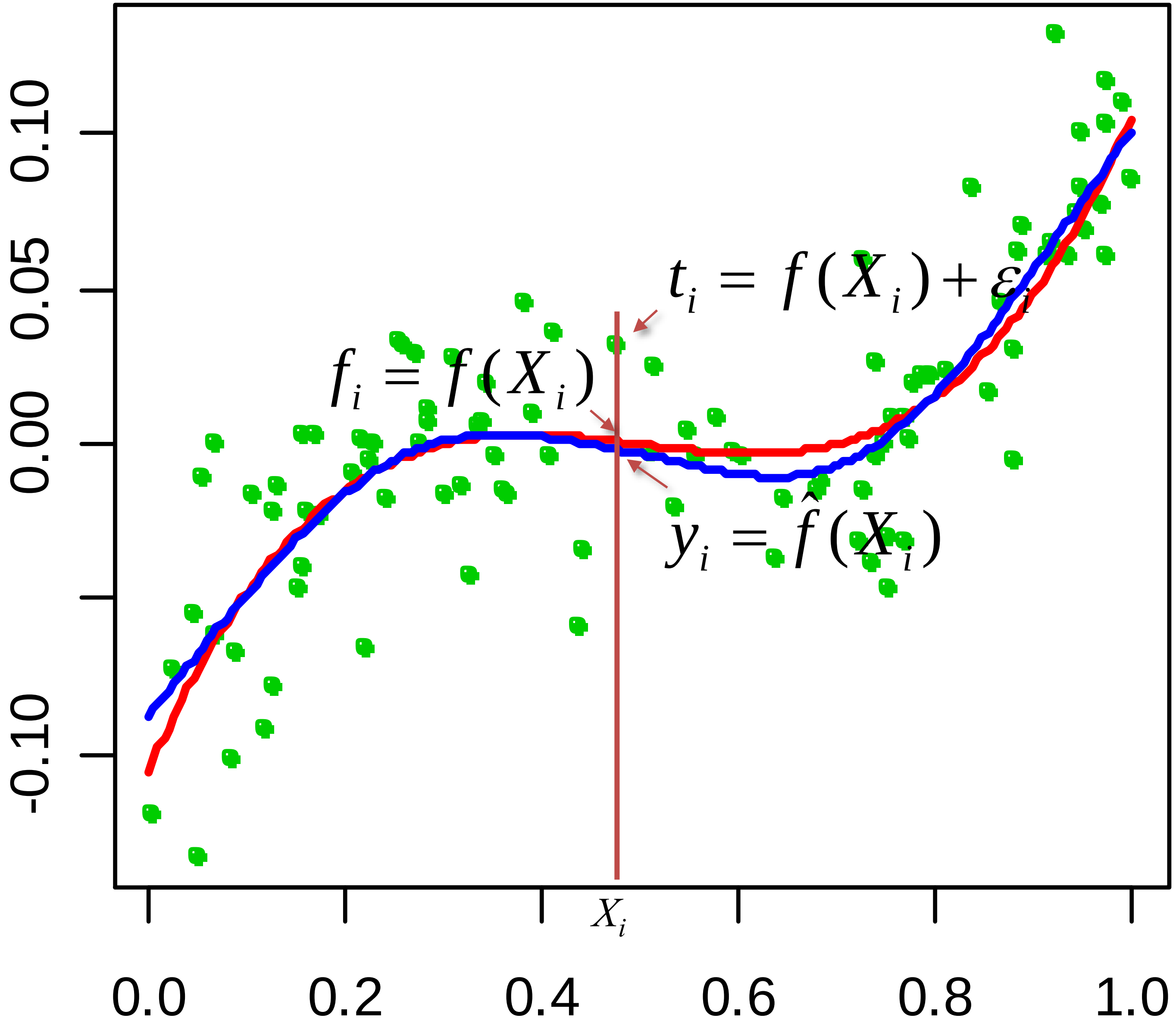
Bias/ Variance Tradeoff

Test vs. Training MSE's illustrates a very important tradeoff that governs the choice of statistical learning methods

- *Bias* refers to the error that is introduced by modeling a real life problem by a much simpler problem
  - E.g., linear regression assumes that there is a linear relationship between Y and X. In real life, some bias will be present
  - The more flexible/complex a method is the less bias it will have

- *Variance* refers to how much your estimate for *f* would change by if you had a different training data set
  - Generally, the more flexible a method is the more variance it has.

New Notation (from ESL)



Bias-Variance in Regression (Part 1)

Let's consider Expected Squared Prediction Error (over any possible data)

$$E\{MSE\} = E\left\{\frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2\right\} = \frac{1}{N}\sum_{i=1}^{N}E\left\{(t_i - y_i)^2\right\}$$

Let apply an "augmentation trick" to the expectation

$$\begin{aligned} E\left\{(t_i - y_i)^2\right\} &= E\left\{(t_i - f_i + f_i - y_i)^2\right\} \\ &= E\left\{(t_i - f_i)^2\right\} + E\left\{(f_i - y_i)^2\right\} + 2E\left\{(f_i - y_i)(t_i - f_i)\right\} \\ &= E\left\{\varepsilon^2\right\} + E\left\{(f_i - y_i)^2\right\} + \cancel{2\left(E\left[f_i t_i\right] - E\left[f_i^2\right] - E\left[y_i t_i\right] + E\left[y_i f_i\right]\right)} \end{aligned}$$

- Being *f* deterministic we have $E\{f_i t_i\} = f_i^2$, $E\{t_i\} = f_i$, and $E\{f_i^2\} = f_i^2$
- Noise is independence $E\{y_i t_i\} = E\{y_i(f_i + \varepsilon)\} = E\{y_i f_i + y_i \varepsilon\} = E\{y_i f_i\} + 0$

Bias-Variance in Regression (Part 2)

From the previous we get something already know

$$E\left\{(t_i - y_i)^2\right\} = E\left\{\varepsilon^2\right\} + E\left\{(f_i - y_i)^2\right\}$$

Lets check the second expected value

$$\begin{aligned}
E\left\{(f_i - y_i)^2\right\} &= E\left\{\left(f_i - E\{y_i\} + E\{y_i\}_i - y_i\right)^2\right\} \\
&= E\left\{\left(f_i - E\{y_i\}\right)^2\right\} + E\left\{\left(E\{y_i\} - y_i\right)^2\right\} + 2E\left\{\left(E\{y_i\} - y_i\right)\left(f_i - E\{y_i\}\right)\right\} \\
&= bias^2 + Var\{y_i\} + \cancel{2\left(E\left[f_i E[y_i]\right] - E\left[E[y_i]^2\right] - E[y_i f_i]_i + E\left[y_i E[y_i]\right]\right)}
\end{aligned}$$

Because *f* is deterministic and $E\{E\{z\}\} = E\{z\}$

$$E\{y_i f_i\} = f_i E\{y_i\} \qquad E\{y_i E\{y_i\}\} = E\{y_i\}^2$$

$$E\left\{E\{y_i\}^2\right\} = E\{y_i\}^2 \qquad E\{f_i E\{y_i\}\} = f_i E\{y_i\}$$

$$\boxed{\begin{gathered}
E\left\{(f_i - y_i)^2\right\} = bias^2 + Var\{y_i\} \\
E\left\{(t_i - y_i)^2\right\} = Var\{noise\} + bias^2 + Var\{y_i\}
\end{gathered}}$$

The Trade-off

For any given, X=*x*, the expected test MSE for a new Y will be

$$\underbrace{E\left\{\left(t_i - y_i\right)^2\right\}}_{\text{Expected Prediction Error}} = \overbrace{Var\{noise\}}^{\text{Irreducible Error}} + \underbrace{bias^2}_{\text{Model Bias}} + \overbrace{Var\{y_i\}}^{\text{Model Variance}}$$

I.e., as a method/model gets more complex

- Bias will decrease
- Variance will increase
- Expected Prediction Error may go up or down!

Test MSE, Bias and Variance

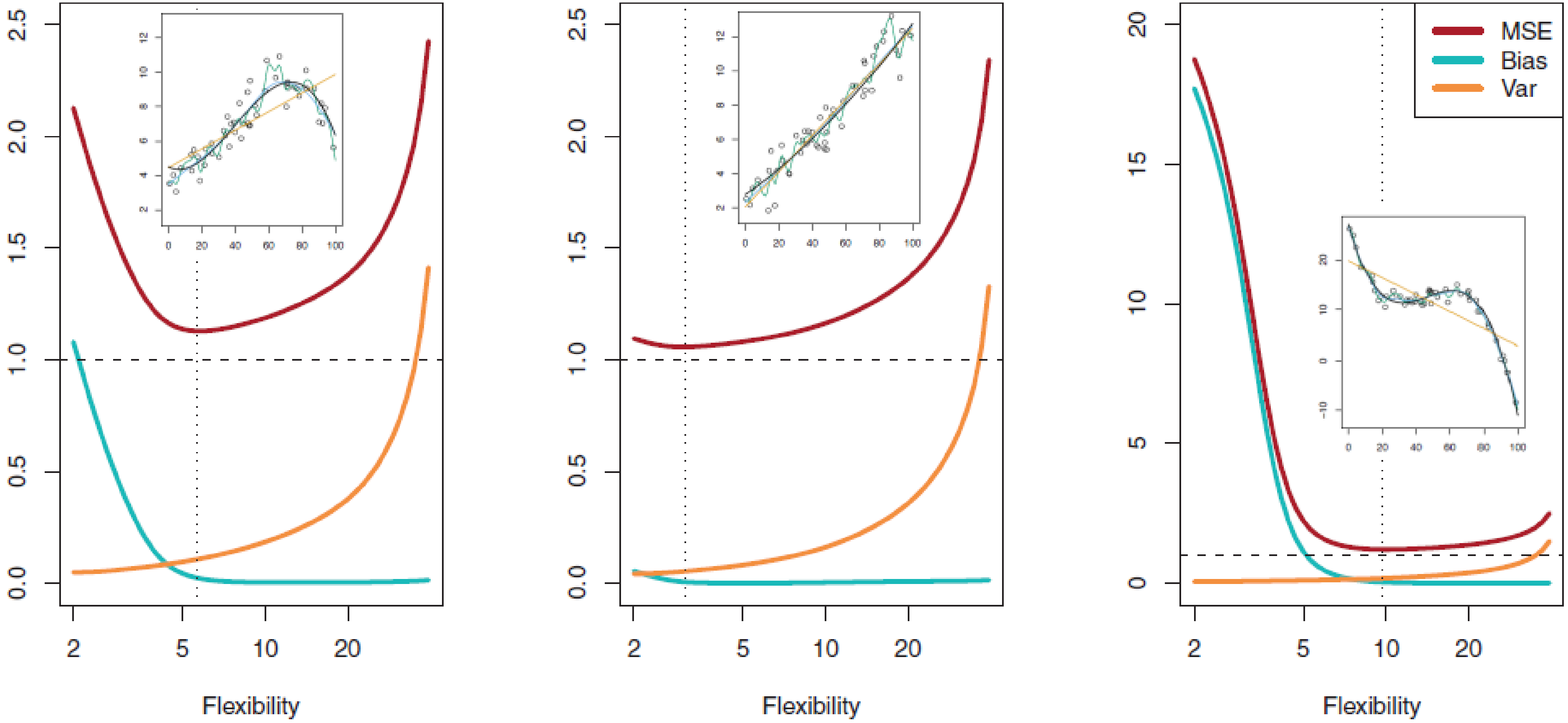


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Can we actually compute those?

For a Linear Model

$$\mathrm{Err}(x_0) = \mathrm{E}[(Y - \hat{f}_\lambda)^2 | X = x_0]$$

$$\sigma^2 + \Big[f(x_0) - \mathrm{E}\hat{f}(x_i)\Big]^2 + \|\mathbf{h}(x_0)\|^2\sigma^2$$

$$\frac{1}{N}\sum_{i=1}^{N}\mathrm{Err}(\mathrm{x_i}) = \sigma^2 + \frac{1}{N}\sum_{i=1}^{N}[f(x_i) - \mathrm{E}\hat{f}(x_i)]^2 \boxed{+\frac{p}{N}\sigma^2}$$

For the KNN regression fit

$$\mathrm{Err}(x_0) = \mathrm{E}[(Y - \hat{f}_\lambda)^2 | X = x_0]$$

$$= \sigma^2 + \left[f(x_0) - \frac{1}{k}\sum_{l=1}^{k} f(x_l)\right]^2 \boxed{+\frac{\sigma^2}{k}}$$

What about Classification?

For a classification problem we can use the error rate i.e.

$$Error\ Rate = \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) / n$$

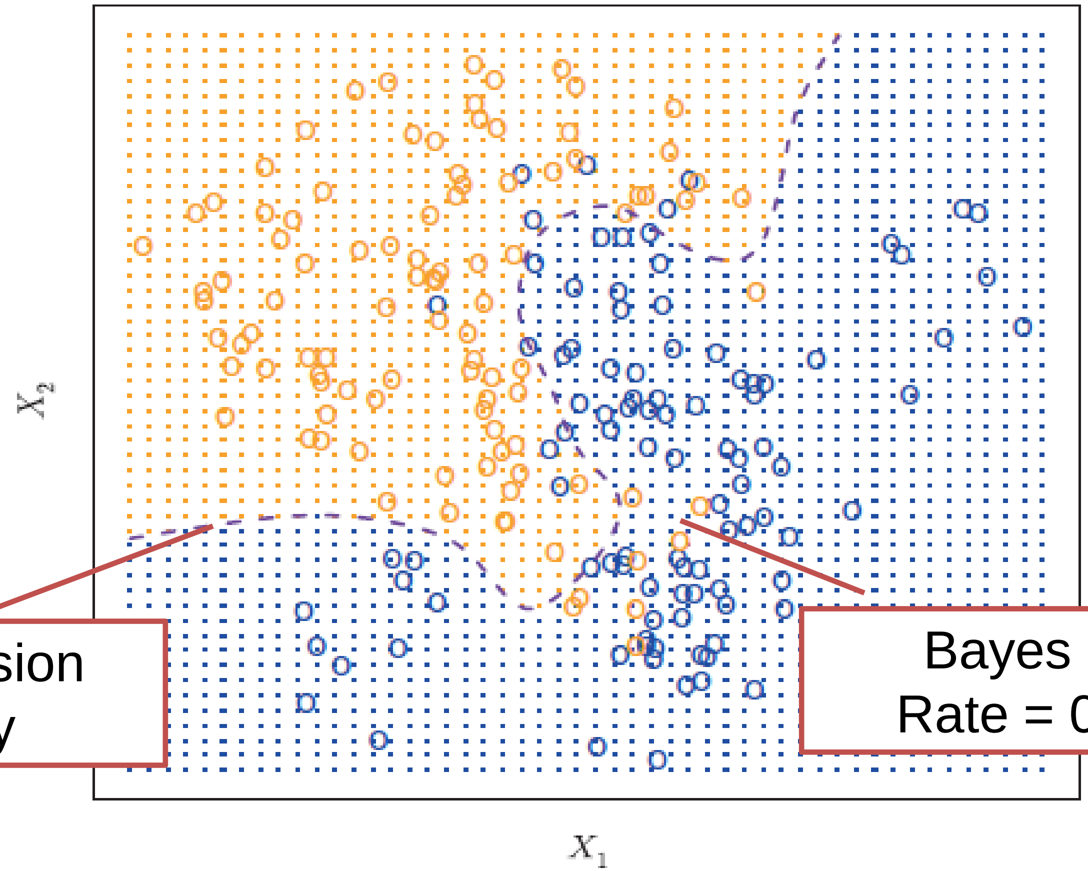
- Where $I(y_i \neq \hat{y}_i)$ is an indicator function, which will give 1 if the condition is correct, otherwise it gives a 0.
- Error rate represents the fraction of incorrect classifications, or misclassifications

The Bayes Classifier minimizes the Average Test Error Rate

$$\max_j P(Y = j \mid X = x_0)$$

The Bayes error rate is the lowest possible Error Rate achievable knowing the "true" distribution of the data: $1 - E\left(\max_j \Pr(Y = j|X)\right)$

Bayes Classifier



Bayes Decision Boundary

Bayes Error Rate = 0.1304

FIGURE 2.13. *A simulated data set consisting of* 100 *observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.*

K-Nearest Neighbors (KNN)

The k Nearest Neighbors method is a non parametric model often used to estimate the Bayes Classifier

- For any given X we find the k closest neighbors to X in the training data, and examine their corresponding Y
- If the majority of the Y's are orange we predict orange otherwise guess blue.

Some notes about such a simple classifier …

- The smaller the k, the more flexible the method will be
- KNN has "zero" training time, some cost at runtime to find the k closest neighbors reduced by indexing
- KNN has problems in high dimensional spaces, it needs approximate methods

KNN Example with k = 3

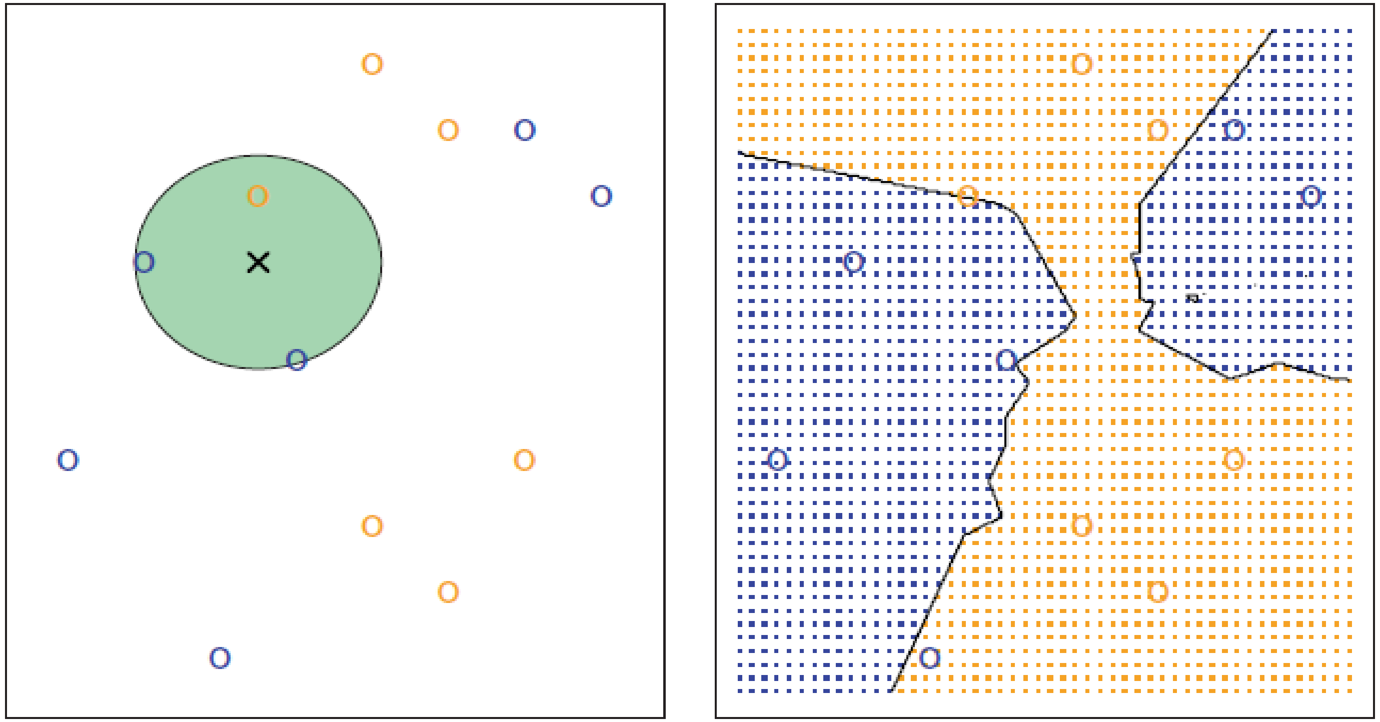
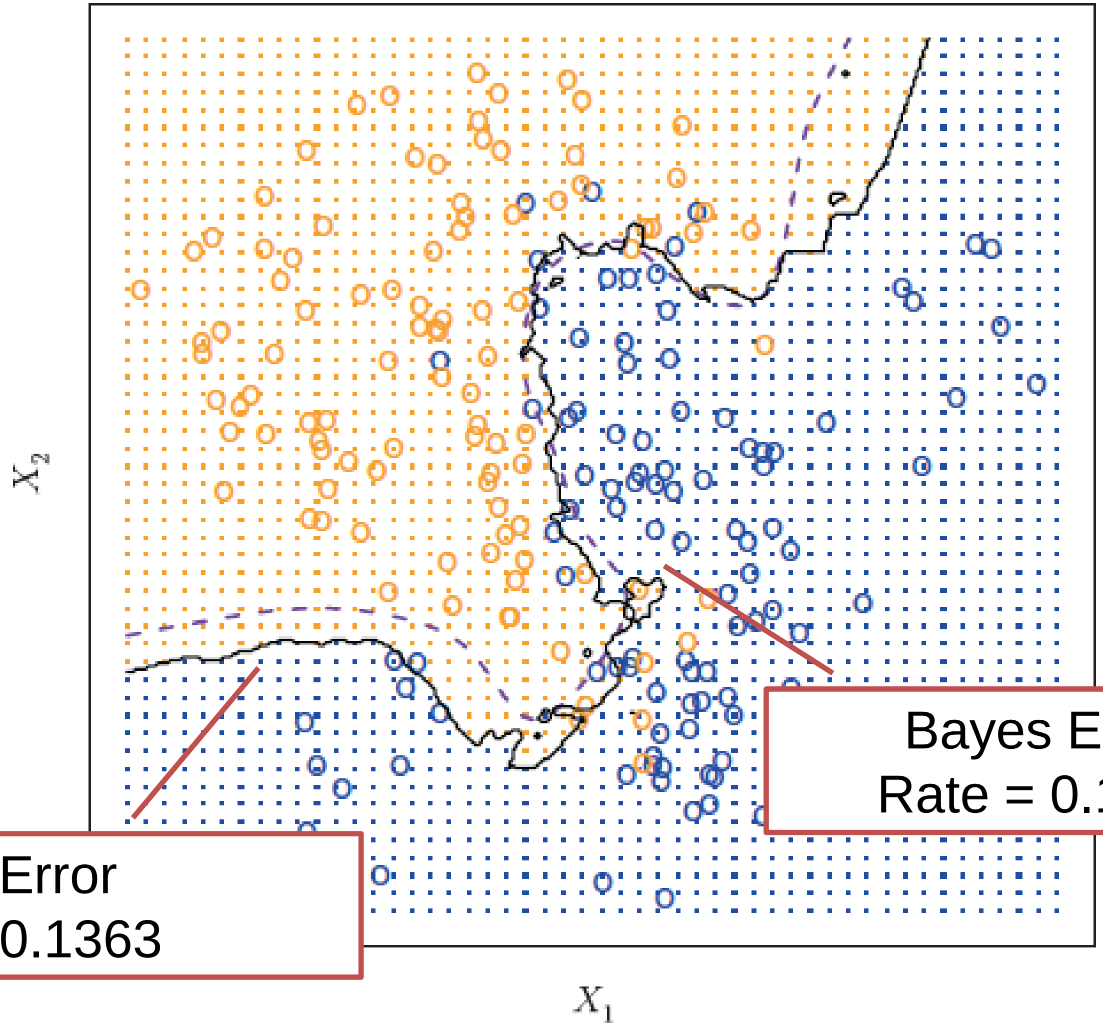


FIGURE 2.14. *The KNN approach, using* $K = 3$, *is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

Simulated Data: K = 10



KNN Error
Rate = 0.1363

Bayes Error
Rate = 0.1304

FIGURE 2.15. *The black curve indicates the KNN decision boundary on the data from Figure 2.13, using $K = 10$. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.*

K = 1 and K = 100

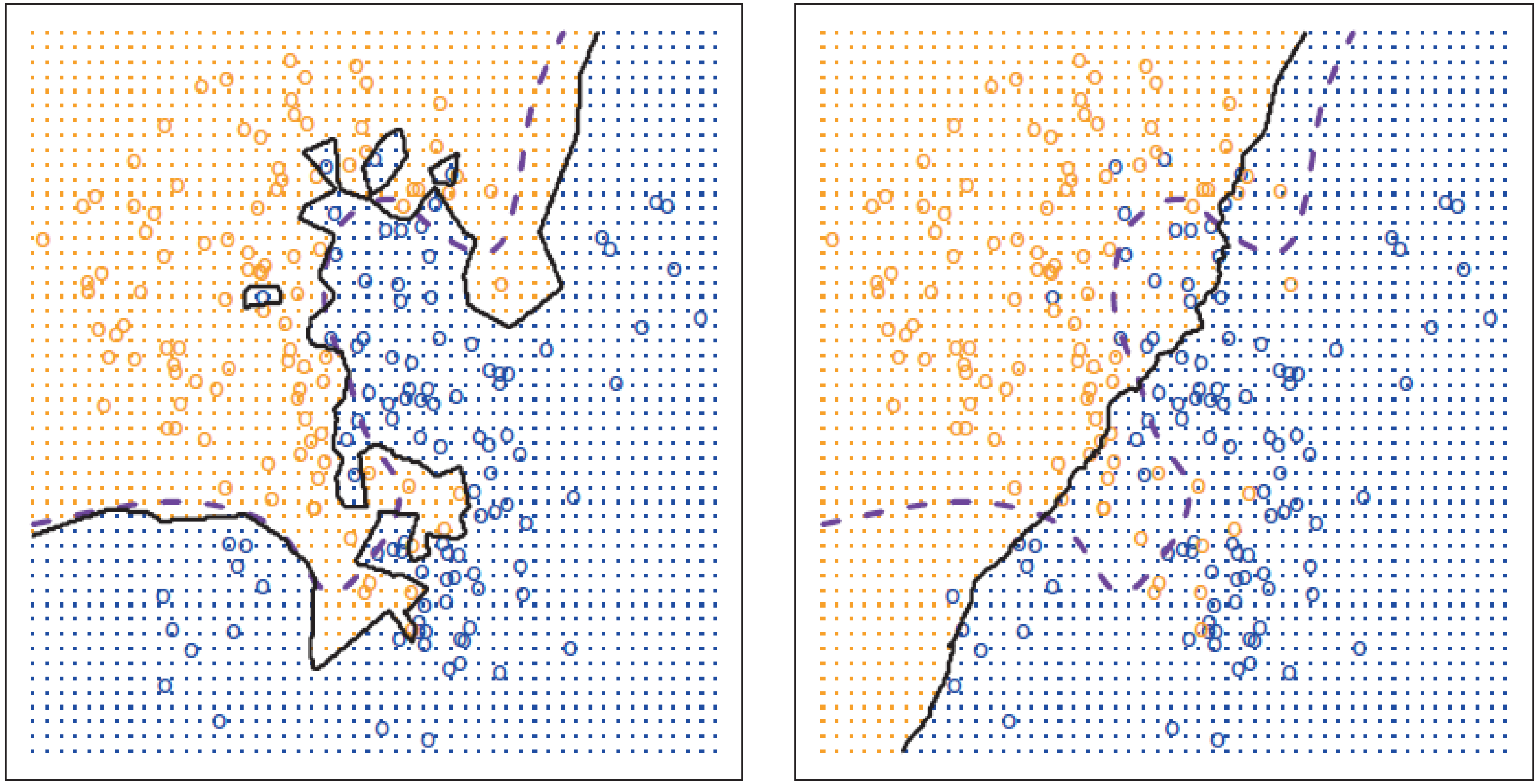


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

Training vs. Test Error Rates

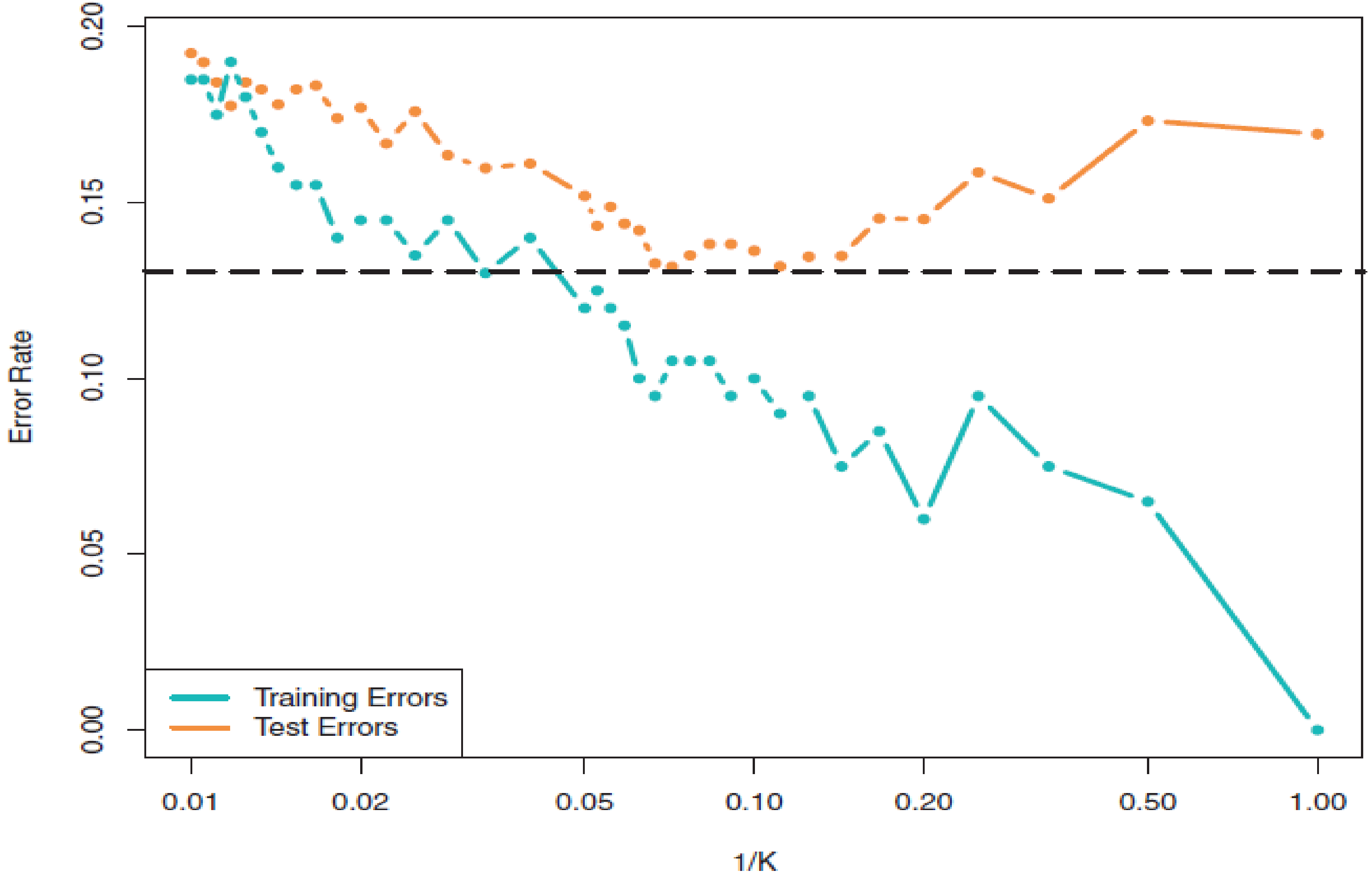
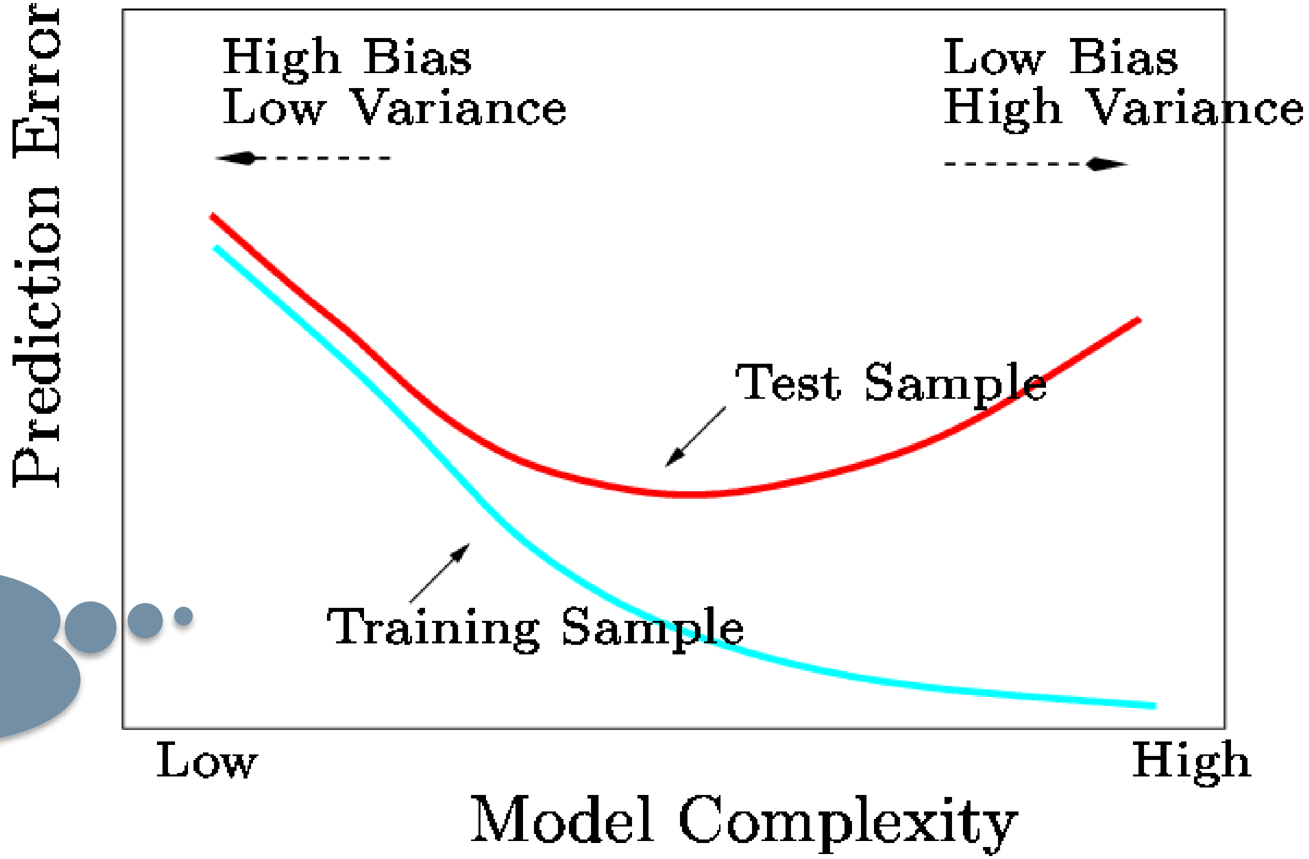


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

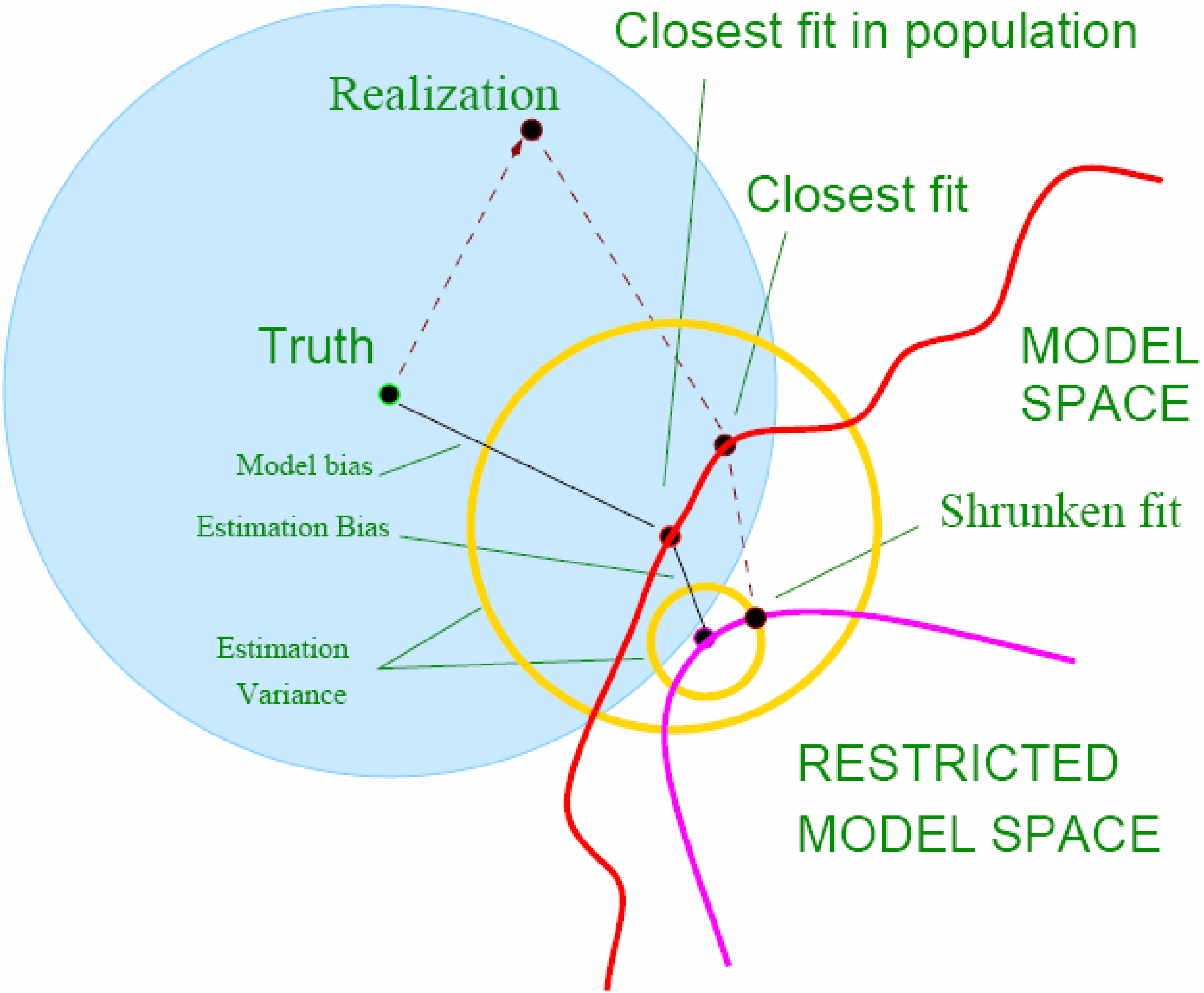
A Fundamental Picture

Training errors will decline while test errors will decline at first (as reductions in bias dominate) but will then start to increase again (as increases in variance dominate).



Overfitting boils down to this!!!

A More Fundamental Picture



Question Time!

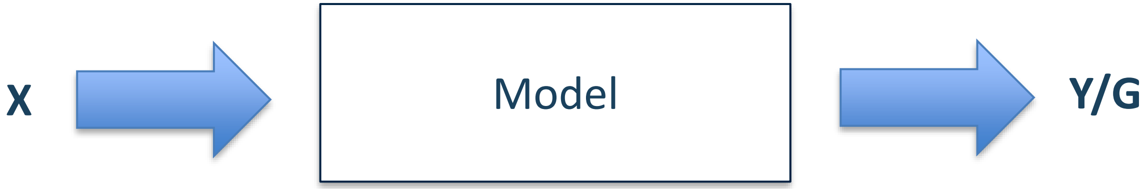
What is Statistical Learning?

Why do we estimate f?

How do we estimate f?

What does the bias-variance trade-off state?

What about classification?



Some important taxonomies … you should by heart!

- Prediction vs. Inference
- Parametric vs. Non Parametric models
- Regression vs. Classification problems
- Supervised vs. Unsupervised learning